

CERTIFIED COPY OF  
PRIORITY DOCUMENT

日 本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

JC882 U.S. PTO  
09/736357



別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日  
Date of Application:

2000年 3月10日

願 番 号  
Application Number:

特願2000-067789

願 人  
Applicant(s):

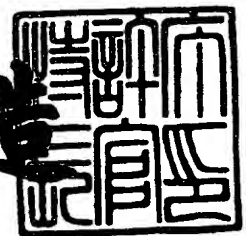
富士通株式会社

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2000年10月20日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



【書類名】 特許願

【整理番号】 9941012

【提出日】 平成12年 3月10日

【あて先】 特許庁長官 近藤 隆彦 殿

【国際特許分類】 G06F 9/312

【発明の名称】 計算機とその制御方法

【請求項の数】 4

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 三宅 英雄

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 須賀 敦浩

【発明者】

【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

【氏名】 中村 泰基

【特許出願人】

【識別番号】 000005223

【氏名又は名称】 富士通株式会社

【代理人】

【識別番号】 100070150

【住所又は居所】 東京都渋谷区恵比寿4丁目20番3号 恵比寿ガーデンプレイスタワー32階

【弁理士】

【氏名又は名称】 伊東 忠彦

【電話番号】 03-5424-2511

【手数料の表示】

【予納台帳番号】 002989

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9704678

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 計算機とその制御方法

【特許請求の範囲】

【請求項 1】 プログラムされた命令の実行を制御する計算機の制御方法であって、

前記プログラムにおいて分岐命令より後置された命令を、前記分岐命令より先に実行する第一のステップと、

前記第一のステップにおいて、前記プログラムの実行を中断する必要性が見出された場合に、前記中断を留保する第二のステップと、

前記分岐命令を実行することにより選択された分岐先の命令の実行において、前記第二のステップで留保した前記中断が必要とされる場合には、前記中断を遂行する第三のステップとを有することを特徴とする計算機の制御方法。

【請求項 2】 前記第三のステップで前記中断を遂行した場合には割り込み処理プログラムを実行する第四のステップをさらに有する請求項 1 に記載の計算機の制御方法。

【請求項 3】 プログラムされた命令を実行する計算機であって、

分岐命令より後置された所定の命令を前記分岐命令より先に実行したとき、前記プログラムの実行を中断する必要性が見出された場合に前記中断を留保する例外抑止手段と、

前記分岐命令を実行することにより選択された分岐先の命令の実行において、前記例外抑止手段により留保された前記中断が必要とされる場合には、前記中断を遂行する中断処理実行手段とを備えたことを特徴とする計算機。

【請求項 4】 前記中断処理実行手段により前記中断が遂行された場合には割り込み処理プログラムを実行する割り込み処理実行手段をさらに備えた請求項 3 に記載の計算機。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、プログラムされた命令を実行する計算機とその制御方法に関するも

のである。

【0 0 0 2】

【従来の技術】

従来より、プログラムの開発におけるデバッグを支援するシステムとしてデバッグが作られているが、このようなシステムにおいては、プログラムの実行を所定の命令により中断するブレークポイント機能が備えられている。

【0 0 0 3】

ここで、上記ブレークポイント機能とは、デバッグ対象プログラムにおいて、予め指定した任意の命令に対して、その都度実行を停止させる機能をいう。そして、このような機能は、プログラムのデバッグ処理において有益であることから、上記のように多くのデバッグに組み込まれている。

【0 0 0 4】

利用者は、このブレークポイント機能を用いる場合、予め実行を停止させたい命令あるいは命令の種類を指定し、デバッグ対象プログラムを起動させる。これにより、デバッグ対象プログラムは、指定された命令で実行を停止する。その際、利用者は必要に応じて、停止した時点でのレジスタやメモリの値を調べたり、変更したりすることができる。なおこの時には、その後において、停止した時点から該プログラムの実行を再開することもできる。

【0 0 0 5】

ここで、上記ブレークポイント機能の実現方法としては、命令ブレークと、ソフトウェアブレークと、データブレークとがある。このうち、命令ブレークにおいては、実行を停止させたい命令のアドレスを保持するブレークポイントレジスタがハードウェアに設けられる。そして、このブレークポイントレジスタで指定される命令が検出されると、割り込みが発生し、割り込み処理プログラムを経由してデバッグ支援プログラムに制御が渡り、利用者にその旨が通知される。そしてさらに実行の再開が指示されると、制御が元に戻される。

【0 0 0 6】

一方、ソフトウェアブレークにおいては、予め指定された命令が割り込みを発生する命令で置き換えられる。そして、該割り込みを発生する命令が検出される

と、割り込みが発生し、割り込み処理プログラムを経由してデバッグ支援プログラムに制御が渡り、利用者にその旨が通知される。そしてさらに実行の再開が指示されると、置き換えられた元の命令を模擬的に実行した後、制御が元に戻される。なおこの方法においては、割り込みを発生する命令として、ブレークポイント命令と呼ばれる専用の命令を用意する方法もある。

## 【 0 0 0 7 】

また、上記のデータブレークにおいては、命令の実行において参照あるいは変更の対象とするデータのメモリアドレス（あるいはメモリアドレス域）を保持するブレークポイントレジスタがハードウェアに設けられる。そして、このレジスタに保持されたメモリアドレス（あるいはメモリアドレス域）を参照あるいは変更する命令が検出されると、割り込みが発生し、割り込み処理プログラムを経由してデバッグ支援プログラムに制御が渡り、利用者にその旨が通知される。そして、その後に実行の再開が指示されると、制御が元に戻される。

## 【 0 0 0 8 】

以下において、上記のような機能を有する従来の計算機について説明する。図 1 は、該従来の計算機の構成を示す図である。図 1 に示されるように、この計算機はメモリ 1 と、メモリ 1 に接続された命令読み出し部 3 0 3 と、メモリ 1 及び命令読み出し部 3 0 3 に接続された命令実行部 3 0 7 と、命令実行部 3 0 7 に接続されたレジスタ制御部 3 0 9 と、命令読み出し部 3 0 3 と命令実行部 3 0 7 及びレジスタ制御部 3 0 9 に接続された割り込み制御部 9 とを備える。

## 【 0 0 0 9 】

ここで、命令読み出し部 3 0 3 は命令読み出し制御部 1 1 と、プログラムカウンタ（PC） 1 3 と、命令語レジスタ（IR） 1 5 と、命令ブレーク検出部 3 0 1 とを含む。そして、命令読み出し制御部 1 1 はメモリ 1 に接続され、プログラムカウンタ（PC） 1 3 は命令読み出し制御部 1 1 に接続される。さらに、命令語レジスタ（IR） 1 5 は命令読み出し制御部 1 1 に接続され、命令ブレーク検出部 3 0 1 はメモリ 1 と後述する命令実行回路 2 3 に接続される。

## 【 0 0 1 0 】

また、命令実行部 3 0 7 は命令デコード部 1 7 と、ロード命令実行部 1 9 と、

ストア命令実行部 2 1 と、演算命令実行部 2 2 と、命令実行回路 2 3 と、データブレーク検出部 3 0 5 とを含む。

## 【 0 0 1 1 】

そして、命令デコード部 1 7 は命令語レジスタ 1 5 に接続され、ロード命令実行部 1 9 はメモリ 1 及び命令デコード部 1 7 と、後述する汎用レジスタ 3 7 に接続される。また、ストア命令実行部 2 1 は命令デコード部 1 7 及び汎用レジスタ 3 7 に接続され、演算命令実行部 2 2 は命令デコード部 1 7 と汎用レジスタ 3 7 に接続され、命令実行回路 2 3 は命令デコード部 1 7 と汎用レジスタ 3 7 及び後述するレジスタ 3 1, 3 3, 3 5 に接続される。さらに、データブレーク検出部 3 0 5 は、ロード命令実行部 1 9 とストア命令実行部 2 1 及び命令実行回路 2 3 に接続される。

## 【 0 0 1 2 】

一方、レジスタ制御部 3 0 9 は、レジスタ (EPCR) 3 1 と、レジスタ (EPSR) 3 3 と、レジスタ (PSR) 3 5 と、汎用レジスタ 3 7 とを含む。ここで、レジスタ (EPCR) 3 1 とレジスタ (EPSR) 3 3 及びレジスタ (PSR) 3 5 は、さらに割込制御回路 4 0 に接続される。また、汎用レジスタ 3 7 はロード命令実行部 1 9 とストア命令実行部 2 1、演算命令実行部 2 2、及び命令実行回路 2 3 に接続される。

## 【 0 0 1 3 】

また、割り込み制御部 9 は割込制御回路 4 0 を含む。そして、この割込制御回路 4 0 は命令読み出し制御部 1 1 とプログラムカウンタ 1 3、命令ブレーク検出部 3 0 1、ロード命令実行部 1 9、ストア命令実行部 2 1、演算命令実行部 2 2、命令実行回路 2 3、データブレーク検出部 3 0 5 などに接続される。

## 【 0 0 1 4 】

上記のような構成を有する計算機において、命令読み出し部 3 0 3 はプログラムカウンタ 1 3 が示す命令語をメモリ 1 から読み出し、命令語レジスタ (IR) 1 5 を介して命令実行部 3 0 7 へ供給する。また命令読み出し制御部 1 1 は、命令実行部 3 0 7 あるいは割り込み処理を行う割込制御回路 4 0 より分岐先アドレスが供給された場合には、それをプログラムカウンタ 1 3 へ書き込む。なお命令

読み出し制御部 1 1 は、それ以外の場合には次の命令語を命令実行部 3 0 7 へ供給するため、読み出すべき命令語のアドレスを示すプログラムカウンタ 1 3 をインクリメントする。また、命令語を読み出す際に割り込みを検出した場合には、割込制御回路 4 0 へ割込信号を供給する。

## 【 0 0 1 5 】

また、命令デコード部 1 7 は、命令語レジスタ 1 5 から供給された命令をデコードし、ロード命令の場合にはロード命令実行部 1 9 へ、ストア命令の場合にはストア命令実行部 2 1 へ、演算命令の場合には演算命令実行部 2 2 へ、分岐命令や割り込み復帰命令その他の命令の場合には命令実行回路 2 3 へそれぞれ供給する。

## 【 0 0 1 6 】

また、ロード命令実行部 1 9 はロード命令が供給された場合には、汎用レジスタ 3 7 から読み出された値に基づいて求められた実効アドレスに対応するメモリ 1 の領域からデータを読み出し、その結果を汎用レジスタ 3 7 へ書き込む。なお、ロード命令を実行する際に割り込みを検出した場合には、割込信号を割込制御回路 4 0 へ供給する。

## 【 0 0 1 7 】

同様に、ストア命令実行部 2 1 はストア命令が供給された場合には、汎用レジスタ 3 7 から読み出された値に基づいて求められた実効アドレスに対応する汎用レジスタ 3 7 の領域からデータを読み出し、その結果を実効アドレスに対応するメモリ 1 の領域へ書き込む。なお、ストア命令を実行する際に割り込みを検出した場合には、割込信号を割込制御回路 4 0 へ供給する。

## 【 0 0 1 8 】

また、演算命令実行部 2 2 は演算命令が供給された場合には、汎用レジスタ 3 7 から読み出された値に基づいて求められた実効アドレスに対応する汎用レジスタ 3 7 の領域からデータを読み出し、そのデータに基づき演算し、該演算結果を汎用レジスタ 3 7 へ書き込む。

## 【 0 0 1 9 】

また、命令実行回路 2 3 は、命令デコード部 1 7 から分岐命令が供給された場



合には、分岐成立時に分岐先アドレスをプログラムカウンタ 1 3 へ供給する。さらに、命令デコード部 1 7 から割り込み復帰命令が供給された場合には、割り込み発生前の動作状態を示す E P S R 3 3 の値を P S R 3 5 へ書き込む。そしてこのとき、E P C R 3 1 から復帰先の命令のアドレスを読み出し、それを分岐先アドレスとしてプログラムカウンタ 1 3 へ供給する。

#### 【 0 0 2 0 】

また、命令実行回路 2 3 は、命令デコード部 1 7 からブレークポイント命令が供給された場合には、ソフトウェアブレークが成立した旨を割込制御回路 4 0 へ通知する。また、命令デコード部 1 7 から命令ブレークポイントレジスタ読み出し命令が供給された場合には、後述する命令ブレーク検出部 3 0 1 の命令ブレークポイントレジスタからブレークポイント対象アドレスを読み出し、汎用レジスタ 3 7 へ書き込む。また、命令デコード部 1 7 から命令ブレークポイントレジスタ書込命令が供給された場合には、汎用レジスタ 3 7 から読み出した値に応じたブレークポイント対象アドレスを命令ブレーク検出部 3 0 1 の命令ブレークポイントレジスタへ書き込む。

#### 【 0 0 2 1 】

また同様に、命令デコード部 1 7 からデータブレークポイントレジスタ読み出し命令が供給された場合には、データブレーク検出部 3 0 5 のデータブレークポイントレジスタからブレークポイント対象アドレスを読み出し、汎用レジスタ 3 7 へ書き込む。また、命令デコード部 1 7 からデータブレークポイントレジスタ書込命令が供給された場合には、汎用レジスタ 3 7 から読み出した値に応じたブレークポイント対象アドレスをデータブレーク検出部 3 0 5 のデータブレークポイントレジスタへ書き込む。

#### 【 0 0 2 2 】

なお、上記の種々の命令を実行する際に割り込みを検出した場合には、割込信号を割込制御回路 4 0 へ供給する。

#### 【 0 0 2 3 】

また、E P C R 3 1 は割り込みからの復帰先に該当する命令のアドレスを保持し、そのアドレスは割り込み発生時に設定される。また、P S R 3 5 は動作状態

(state) を示すデータを保持し、E P S R 3 3 は割り込み発生時に設定された割り込み発生前の動作状態を示すデータを保持する。

【 0 0 2 4 】

また、割込制御回路 4 0 は命令読み出し部 3 0 3 あるいは命令実行部 3 0 7 から供給された割込信号に基づいて、割り込みからの復帰先に該当する命令のアドレスを E P C R 3 1 へ、割り込み発生前の動作状態を示すデータを E P S R 3 3 へ、発生した割り込みに対応する動作状態を示す P S R 3 5 へ書き込む。また、発生した割り込みに対応する分岐先アドレスを命令読み出し部 3 0 3 へ供給する。

【 0 0 2 5 】

以上より、上記計算機の動作を要約すると、初期動作状態を通常の動作としたとき次のようになる。通常の動作においては、命令読み出し部 3 0 3 は、プログラムカウンタ 1 3 が示す命令語を読み出し、命令実行部 3 0 7 へ供給し、命令実行部 3 0 7 は供給された命令を実行する。

【 0 0 2 6 】

一方、割り込みが発生した時には、割込制御回路 4 0 は命令読み出し部 3 0 3 あるいは命令実行部 3 0 7 から供給された割込信号に基づいて、割り込みからの復帰先に該当する命令のアドレスを E P C R 3 1 へ、割り込み発生前の動作状態を示すデータを E P S R 3 3 へ、発生した割り込みに対応する動作状態を示す P S R 3 5 へ書き込む。また、発生した割り込みに対応する分岐先アドレスを命令読み出し部 3 0 3 へ供給する。また、命令読み出し部 3 0 3 は割り込み制御部 9 から供給された分岐先アドレスに応じて命令語を読み出し、命令実行部 3 0 7 へ供給する。そして、以降においては上記の通常の場合と同様に動作する。

【 0 0 2 7 】

また、割り込みから復帰する時は、命令実行部 3 0 7 は割り込み復帰命令を実行することにより、E P S R 3 3 の値を P S R 3 5 へ書き込む。また、E P C R 3 1 に格納されたデータを読み出し、その結果を分岐先アドレスとして命令読み出し部 3 0 3 へ供給する。このとき命令読み出し部 3 0 3 は、命令実行部 3 0 7 から供給された分岐先アドレスに基づいて命令語を読み出し、命令実行部 3 0 7

へ供給する。そして、以降においては上記通常の動作がなされる。

#### 【 0 0 2 8 】

次に、図 1 に示された命令ブレーク検出部 3 0 1 について詳しく説明する。図 2 は、上記命令ブレーク検出部 3 0 1 の構成を示す図である。図 2 に示されるように、命令ブレーク検出部 3 0 1 は検出部 3 1 1 ～ 3 1 4 と、アドレスフィールド (ADDR) 3 1 5 ～ 3 1 8 と、E フィールド (E) 3 1 9 ～ 3 2 2 と、V フィールド (V) 3 2 3 ～ 3 2 6 と、OR 回路 3 2 7 とを含む。

#### 【 0 0 2 9 】

ここで、アドレスフィールド (ADDR) 3 1 5 ～ 3 1 8 は、それぞれブレークポイント対象アドレスを保持し、命令ブレークポイントレジスタを構成する。また、E フィールド (E) 3 1 9 ～ 3 2 2 は、命令ブレーク動作が有効か否かを示すデータを保持し、具体的には命令ブレーク動作が無効である場合には 0 を、命令ブレーク動作が有効である場合には 1 を保持する。そして、この E フィールド (E) 3 1 9 ～ 3 2 2 は、命令ブレークポイントレジスタを構成する。また、V フィールド (V) 3 2 3 ～ 3 2 6 は、命令ブレーク検出済みか否かを示すデータを保持し、具体的には命令ブレーク未検出の時には 0 を、命令ブレーク検出済みの時には 1 を保持する。

#### 【 0 0 3 0 】

また、各検出部 3 1 1 ～ 3 1 4 は、命令ブレークが成立するか否かを判定し、より具体的にはメモリ 1 から供給された命令アドレスと命令ブレークポイントレジスタから供給されたアドレスとを比較する。そして、これら両アドレスが一致した場合には、対応する V フィールド (V) 3 2 3 ～ 3 2 6 に 1 を書き込むと共に、一致信号  $m t$  を OR 回路 3 2 7 へ供給する。これにより、この場合には命令ブレークが成立した旨が OR 回路 3 2 7 から割込制御回路 4 0 へ割込信号により通知される。

#### 【 0 0 3 1 】

次に、図 1 に示されたデータブレーク検出部 3 0 5 について詳しく説明する。図 3 は、上記データブレーク検出部 3 0 5 の構成を示す図である。図 3 に示されるように、データブレーク検出部 3 0 5 は検出部 3 1 1 ～ 3 1 4 と、アドレスフ

ィールド (ADDR) 3 1 5 ~ 3 1 8 と、E フィールド (E) 3 1 9 ~ 3 2 2 と、V  
フィールド (V) 3 2 3 ~ 3 2 6 と、OR 回路 3 2 7 とを含む。

#### 【 0 0 3 2 】

ここで、アドレスフィールド (ADDR) 3 1 5 ~ 3 1 8 は、それぞれブレークポ  
イント対象アドレスを保持し、データブレークポイントレジスタを構成する。ま  
た、E フィールド (E) 3 1 9 ~ 3 2 2 は、データブレーク動作が有効か否かを  
示すデータを保持し、具体的にはデータブレーク動作が無効である場合には 0 を  
、データブレーク動作が有効である場合には 1 を保持する。そして、この E フィ  
ールド (E) 3 1 9 ~ 3 2 2 は、データブレークポイントレジスタを構成する。  
また、V フィールド (V) 3 2 3 ~ 3 2 6 は、データブレーク検出済みか否かを  
示すデータを保持し、具体的にはデータブレーク未検出の時には 0 を、データブ  
レーク検出済みの時には 1 を保持する。

#### 【 0 0 3 3 】

また、各検出部 3 1 1 ~ 3 1 4 は、データブレークが成立するか否かを判定し  
、より具体的にはメモリ 1 から供給されたロードストア命令の実効アドレス (デ  
ータアドレス) とアドレスフィールド (ADDR) 3 1 5 ~ 3 1 8 に格納されたブレ  
ークポイント対象アドレスとを比較する。そして、これら両アドレスが一致した  
場合には、対応する V フィールド (V) 3 2 3 ~ 3 2 6 に 1 を書き込むと共に、  
一致信号 *mt* を OR 回路 3 2 7 へ供給する。これにより、この場合にはデータブ  
レークが成立した旨が OR 回路 3 2 7 から割込制御回路 4 0 へ割込信号により通  
知される。

#### 【 0 0 3 4 】

次に、割り込み処理プログラムによる上記計算機のデータブレーク割り込み動  
作の概要を、図 4 のフローチャートを参照しつつ説明する。まず図 4 に示される  
ように、ステップ S 1 ではコンテキストを退避する。次に、ステップ S 2 におい  
てデータブレーク処理を行い、ステップ S 3 でコンテキストの復元がなされる。  
そして、ステップ S 4 において割り込み復帰命令を実行し、割り込み処理から復  
帰して割り込み処理動作を終了する。

#### 【 0 0 3 5 】

次に、割り込み処理プログラムによるソフトウェアブレーク割り込み動作の概要を、図5のフローチャートを参照しつつ説明する。まず図5に示されるように、ステップS1ではコンテキストを退避する。次に、ステップS2においてソフトウェアブレーク処理を行い、ステップS3でコンテキストの復元がなされる。そして、ステップS4において割り込み復帰命令を実行し、割り込み処理から復帰して割り込み処理動作を終了する。

## 【0036】

ところで、近年の計算機では、処理性能を向上すべくコンパイラ他によりプログラムが生成された時に命令の順序を変更するという手法が用いられており、このような手法は命令移動と呼ばれる。そしてこれは、命令実行部の有効稼働率を高めると共に、早期に命令の実行を開始することで、命令のレイテンシを隠蔽すること、すなわち例えば、ロード命令の場合には、メモリからのロード操作を早期に開始することで、潜在的なキャッシュミスを含むメモリへのアクセスの遅延を隠蔽できる可能性があるといったことを狙いとした手法である。

## 【0037】

ここで、上記のような命令移動において、特にコンパイラにおける基本ブロックの境界を越えた命令移動を行うものは、広域命令移動と呼ばれており、さらに広域命令移動において条件分岐を越えて命令移動を行うものは、投機的命令移動と呼ばれている。しかし、このように投機的に移動した命令において例外が発生した場合、例外処理を必要とするため大幅な性能の低下を招いたり、本来のプログラムの順序ではあり得なかったプログラム実行における不正な停止を招いたりする。

## 【0038】

すなわち例えば、必要とされるデータがメモリ上に無い状態であるページフォルト (page fault) を起こす可能性がある命令を投機的に移動した場合には、移動先においてページフォルトを伴う例外処理プログラムが実行されることとなり、本来の順序では生じ得なかったページフォルトによって性能が低下する。また、例えば除算命令を投機的に移動した場合には、例外処理プログラムにおいてゼロで除算する処理 (ゼロ除算) が発生することも生じるが、このような場合には

プログラムの実行が停止することとなる。なお上記のように、投機的に移動した命令の実行において発生した例外は投機的例外と呼ばれる。

#### 【 0 0 3 9 】

そして、上記問題を解決する方法として、例外抑止 (non-excepting) 命令を用いて投機的例外の発生を延期する方法については知られている。すなわち、投機的例外処理を延期する例外抑止命令方式について、文献 "A VLIW architecture for a trace scheduling compiler", Proceedings of the 2nd International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 180-192, 1987 (B. P. Colwel, B. P. Nix, J. J. O'Donnell, D. B. Papworth and P. K. Rodman 共著) が発行されており、例外抑止命令方式を拡張し、投機的例外からの実行の再開を実現したスケジューリングについて、文献 "Sentinel Scheduling for VLIW and Superscalar Processor", Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 238-247, 1992 (S. A. Mahlke, W. Y. Chen, W. W. Hwu, B. R. Rau and M. S. Schlansker 共著) が発行されている。

#### 【 0 0 4 0 】

そして従来においては、上記のような例外抑止命令を含むプログラムのデバッグを行う場合に、投機的に移動した命令のうち本来の順序では確定していない命令によっても、データブレイクによりプログラムの実行が中断してしまうという問題があった。

#### 【 0 0 4 1 】

##### 【発明が解決しようとする課題】

本発明は、上述の問題を解消するためになされたもので、データ処理能力が高く動作の信頼性が向上された計算機とその制御方法を提供することを目的とする。

#### 【 0 0 4 2 】

##### 【課題を解決するための手段】

上記の目的は、プログラムされた命令の実行を制御する計算機の制御方法であ

って、プログラムにおいて分岐命令より後置された命令を、分岐命令より先に実行する第一のステップと、第一のステップにおいて、プログラムの実行を中断する必要性が見出された場合に、中断を留保する第二のステップと、分岐命令を実行することにより選択された分岐先の命令の実行において、第二のステップで留保した中断が必要とされる場合には、中断を遂行する第三のステップとを有することを特徴とする計算機の制御方法を提供することによって達成される。このような手段によれば、分岐命令より後置された命令を、分岐命令より先に実行することによる無用な中断を回避して、プログラムされた命令の実行を確実に遂行することができる。

#### 【0043】

ここで、第三のステップで上記中断を遂行した場合には割り込み処理プログラムを実行する第四のステップをさらに有することにより、該中断によって、例えばデバッグ支援プログラムの実行などによる異なるシステムの制御を実現することができる。

#### 【0044】

また、本発明の目的は、プログラムされた命令を実行する計算機であって、分岐命令より後置された所定の命令を分岐命令より先に実行したとき、プログラムの実行を中断する必要性が見出された場合に上記中断を留保する例外抑止手段と、分岐命令を実行することにより選択された分岐先の命令の実行において、例外抑止手段により留保された中断が必要とされる場合には、中断を遂行する中断処理実行手段とを備えたことを特徴とする計算機を提供することにより達成される。

#### 【0045】

ここで、中断処理実行手段により中断が遂行された場合には割り込み処理プログラムを実行する割り込み処理実行手段をさらに備えたものとすることができる。

#### 【0046】

#### 【発明の実施の形態】

以下において、本発明の実施の形態を図面を参照して詳しく説明する。なお、

図中同一符号は同一又は相当部分を示す。

#### 【実施の形態 1】

図 6 は、本発明の実施の形態 1 に係る計算機の構成を示す図である。図 6 に示されるように、本実施の形態 1 に係る計算機は、図 1 に示された従来の計算機に対してレジスタ制御部 3 3 1 内に例外抑止フラグ (GNER) 3 8 を、命令実行部 3 2 9 内に例外抑止ロード命令実行部 2 4 をさらに備える点で相違する。また、上記例外抑止フラグ (GNER) 3 8 が備えられたことに伴い、演算命令実行部 2 3 5 の構成が図 1 に示された演算命令実行部 2 2 と異なるものとされる。

#### 【0 0 4 7】

すなわち図 7 に示されるように、演算命令実行部 2 3 5 は、演算制御回路 2 4 7 と、フラグ検出回路 2 4 3 と、セレクタ 2 4 5 と、OR 回路 2 4 8 とを含む。ここで、演算制御回路 2 4 7 の入力端は命令デコード部 1 7 及び汎用レジスタ 3 7 に接続され、出力端は OR 回路 2 4 8 を介して割込制御回路 4 0 と接続され、さらに汎用レジスタ 3 7 とセレクタ 2 4 5 に接続される。また、フラグ検出回路 2 4 3 の入力端は汎用レジスタ 3 7 に接続され、出力端はセレクタ 2 4 5 及び OR 回路 2 4 8 に接続される。そして、セレクタ 2 4 5 の出力端は汎用レジスタ 3 7 に接続される。

#### 【0 0 4 8】

また、図 6 に示されるように、例外抑止ロード命令実行部 2 4 はメモリ 1 と、命令デコード部 1 7 及び汎用レジスタ 3 7 に接続され、例外抑止フラグ 3 8 は汎用レジスタ 3 7 に並設される。

#### 【0 0 4 9】

上記のような構成を有する本実施の形態 1 に係る計算機において、例外抑止ロード命令実行部 2 4 は、命令デコード部 1 7 から例外抑止ロード命令が供給された場合には、まず最初に汎用レジスタ 3 7 より読み出した値から実効アドレスを求め、該実効アドレスに対応するメモリ 1 内の領域からデータを読み出す。そして、データを読み出す際にデータブレイク等の例外要因が存在するか否か確認する。

#### 【0 0 5 0】



その結果、該例外要因が存在しないと判断された場合には、メモリ 1 内のロード対象アドレス域から読み出した上記データを、汎用レジスタ 3 7 へ書き込む。

## 【 0 0 5 1 】

一方、該例外要因が存在すると判断された場合には、検出された該例外要因を識別する識別値が汎用レジスタ 3 7 に記憶される。このとき、上記識別値が記憶されるレジスタに対応する例外抑止フラグ 3 8 に 1 の値が設定される。なお、上記例外要因を識別する識別値の例は、次の表 1 に示される。

## 【 0 0 5 2 】

【表 1】

例外要因	識別値
0 除算	0 x 2 8
データアクセスエラー	0 x 3 2

また、上記の例外抑止フラグ 3 8 は、その各ビットが汎用レジスタ 3 7 の各エントリに対応し、例外要因が未検出の状態では 0 とされ、例外抑止ロード命令の実行において例外要因が検出された場合には 1 とされる。

## 【 0 0 5 3 】

一方、本実施の形態に係る計算機においては、メモリ 1 内の所定のアドレス領域に、例外抑止ロード命令テーブルと、コミットポイントテーブル、コミットブレークポイントテーブル、及び例外抑止データブレーク履歴テーブルが格納される。

## 【 0 0 5 4 】

ここで例外抑止ロード命令テーブルは、図 1 2 に示されるように、例外抑止ロード命令の命令アドレスと、該例外抑止ロード命令が含まれる制御経路（制御パス）の識別番号とを対にしたデータ、すなわち（a 0, p 0）、（a 1, p 1）、…（a i, p i）から構成される。また、同様にコミットポイントテーブルは、図 1 3 に示されるように、制御経路の識別番号と、その制御経路が本来の順序による実行において確定する場所を示す命令アドレスと、該制御経路が確定する

ことで無効となる他の制御経路の識別番号とを組にしたデータ、すなわち  $(p_0, b_0, sp_0)$ 、 $(p_1, b_1, sp_1)$ 、 $\dots (p_j, b_j, sp_j)$  から構成される。

## 【0055】

また、コミットブレークポイントテーブルは、図14に示されるように、制御経路が本来の順序による実行において確定する場所を示す命令アドレスと、該命令アドレスに対応する制御経路の識別番号とを組にしたデータ、すなわち  $(b_0, p_0)$ 、 $(b_1, p_1)$ 、 $\dots (b_k, p_k)$  から構成される。また、例外抑止データブレーク履歴テーブルは、図15に示されるように、ブレーク対象とされる命令が含まれる制御経路の識別番号と、該ブレーク対象とされる命令の命令アドレスと、上記命令の実効アドレスとを組にしたデータ、すなわち  $(p_0, c_0, ec_0)$ 、 $(p_1, c_1, ec_1)$ 、 $\dots (p_l, c_l, ec_l)$  から構成される。

## 【0056】

次に、上記のような構成を有する計算機の動作を説明する。なお、本実施の形態に係る計算機における以下の動作は、プログラムされた命令からなる所定のソフトウェアを実行することにより実現される。

## 【0057】

図8は、図6に示された計算機の、データブレークを検出した際の動作を示すフローチャートである。図8に示されるように、まずステップS1で、命令ブレーク検出部301によりブレーク対象とする命令が検出された場合には、該命令が例外抑止命令であるか否かが、演算命令実行部235の命令比較動作により判定される。そして、ステップS2において該命令が例外抑止命令であると判断された場合には、ステップS3へ進み、例外抑止命令でないと判断された場合にはステップS10へ進む。

## 【0058】

このとき、ステップS3では該ブレーク対象命令に関する情報を、メモリ1内に格納された上記例外抑止データブレーク履歴テーブルに登録し、動作を終了する。一方、ステップS10ではデータブレーク処理を実行して、動作を終了する。

## 【 0 0 5 9 】

図 9 は、図 6 に示された計算機において、分岐命令の実行により本来の順序での命令の実行が確定したときの動作を示すフローチャートである。図 9 に示されるように、まずステップ S 1 では、実行が確定した制御経路のデータブレークがメモリ 1 に格納されている例外抑止データブレーク履歴テーブルに存在するものであるか否か確認する。そしてステップ S 2 において、演算命令実行部 2 3 5 の比較動作により、該制御経路のデータブレークがメモリ 1 に格納されている例外抑止データブレーク履歴テーブルに存在すると判断された場合にはステップ S 1 0 へ進み、存在しないと判断された場合にはステップ S 3 へ進む。

## 【 0 0 6 0 】

そしてステップ S 3 では、他の制御経路におけるデータブレークに関する情報が、上記例外抑止データブレーク履歴テーブル内に存在するか否かがサーチされ、ステップ S 4 で演算命令実行部 2 3 5 により該情報が存在するものと判断された場合にはステップ S 5 へ進み、存在しないものと判断された場合には動作を終了する。ここで、ステップ S 5 では例外抑止データブレーク履歴テーブルに含まれる上記他の制御経路におけるデータブレークに関する情報を無効化し、動作を終了する。

## 【 0 0 6 1 】

一方、ステップ S 1 0 では、他の制御経路におけるデータブレークに関する情報が、上記例外抑止データブレーク履歴テーブル内に存在するか否かがサーチされ、ステップ S 1 1 において演算命令実行部 2 3 5 により存在するものと判断された場合にはステップ S 1 2 へ進み、存在しないものと判断された場合にはステップ S 1 3 へ進む。ここで、ステップ S 1 2 では例外抑止データブレーク履歴テーブルに含まれる上記他の制御経路におけるデータブレークに関する情報を無効化する。

## 【 0 0 6 2 】

そして、ステップ S 1 3 では、ステップ S 2 において見出されたデータブレークの処理が実行され、ステップ S 1 4 では例外抑止データブレーク履歴テーブル

に含まれる実行が確定した制御経路のデータブレイク、すなわちステップ S 1 3 で実行した処理、に関する情報を無効化して動作を終了する。

#### 【 0 0 6 3 】

次に図 6 に示された計算機の、割込み処理プログラムによるデータブレイク割込み動作を、図 1 0 のフローチャートを参照しつつ説明する。図 1 0 に示されるように、まずステップ S 1 ではコンテキストが退避される。そして次に、ステップ S 2 においてブレイク対象命令を読み出し、該命令が例外抑止命令であるか否かが、演算命令実行部 2 3 5 の命令比較動作により判定される。そして、ステップ S 3 において該命令が例外抑止命令であると判断された場合には、ステップ S 4 へ進み、例外抑止命令でないと判断された場合にはステップ S 2 0 へ進む。

#### 【 0 0 6 4 】

このとき、ステップ S 4 では例外抑止ロード命令テーブルを参照し、該ブレイク対象命令が含まれる制御経路番号を求める。そして、ステップ S 5 において、メモリ 1 内に格納された上記例外抑止データブレイク履歴テーブルに、ブレイク対象命令が含まれる上記制御経路番号と、命令アドレスと、実効アドレスとを登録する。次に、ステップ S 6 において、上記制御経路番号に基づき、メモリ 1 に格納されたコミットポイントテーブルを参照し、他の制御経路が存在するか否かを確認する。そして、ステップ S 7 において演算命令実行部 2 3 5 の比較動作により他の制御経路が存在すると判断された場合にはステップ S 8 へ進み、存在しないと判断された場合にはステップ S 1 1 へ進む。

#### 【 0 0 6 5 】

ここでステップ S 8 では、見出された制御経路番号が求められ、ステップ S 9 ではコミットブレイクポイントテーブルへ、該他の制御経路のコミットポイントに対応するブレイクポイントの命令アドレスと、制御経路番号とを登録する。このようにして、ステップ S 1 0 において該他の制御経路のコミットポイントに対するブレイクポイントを設定する。

#### 【 0 0 6 6 】

そして、ステップ S 1 1 において、ステップ S 1 で退避されたコンテキストが復元される。なお、ステップ S 2 0 ではデータブレイク処理を実行し、上記ステ

ップS11へ動作が進行する。そしてさらにステップS12においては、割り込み復帰命令を実行することにより、データブレーク割込みの処理から元のプログラムの実行へ復帰し、割込み処理動作を終了する。

#### 【0067】

次に、図6に示された計算機の、割込み処理プログラムによるソフトウェアブレーク割込み動作を、図11に示されたフローチャートを参照しつつ説明する。図11に示されるように、まずステップS1ではコンテキストが退避される。そして次に、ステップS2においてブレーク対象命令の命令アドレスがコミットブレークポイントテーブルに存在するか否かが確認され、ステップS3において演算命令実行部235の比較動作により該命令アドレスが存在すると判断された場合には、ステップS4へ進み、存在しないと判断された場合にはステップS40へ進む。

#### 【0068】

そして、ステップS4ではコミットブレークポイントテーブルからブレーク対象命令の命令アドレスに対応する制御経路番号を求める。次に、ステップS5において、例外抑止データブレーク履歴テーブルに上記の制御経路番号が存在するか否かが演算命令実行部235の比較動作によりサーチされ、ステップS6において存在すると判断された場合にはステップS20へ進み、存在しないと判断された場合にはステップS7へ進む。そして、ステップS7ではコミットポイントテーブルに他の制御経路が存在するか否かが演算命令実行部235の比較動作によりサーチされ、ステップS8において存在すると判断された場合にはステップS9へ進み、存在しないと判断された場合にはステップS13へ進む。

#### 【0069】

ステップS9では、ステップS8において見出された他の制御経路のコミットポイントにおける命令アドレスを求める。そして、ステップS10では上記命令アドレスに対するブレークポイントを、元の命令を復元することにより解除する。次に、ステップS11では、コミットブレークポイントテーブルから上記他の制御経路の制御経路番号と一致するエントリを無効化する。そしてさらに、ステップS12では、例外抑止データブレーク履歴テーブルにおいて、該他の制御経

路の制御経路番号に一致するエントリを無効化する。

【 0 0 7 0 】

次に、ステップ S 1 3 ではブレーク対象命令の命令アドレスに対するブレークポイントを解除し、ステップ S 1 4 においてコンテキストが復元される。そして最後に、ステップ S 1 5 において割り込み復帰命令を実行することにより割り込み処理から復帰し、割り込み処理動作を終了する。

【 0 0 7 1 】

一方、ステップ S 2 0 では、コミットポイントテーブルに他の制御経路が存在するか否かが演算命令実行部 2 3 5 の比較動作によりサーチされ、ステップ S 2 1 において存在すると判断された場合にはステップ S 2 2 へ進み、存在しないと判断された場合にはステップ S 2 6 へ進む。

【 0 0 7 2 】

ステップ S 2 2 では、ステップ S 2 1 において見出された他の制御経路のコミットポイントにおける命令アドレスを求める。そして、ステップ S 2 3 では上記命令アドレスに対するブレークポイントを、元の命令を復元することにより解除する。次に、ステップ S 2 4 では、コミットブレークポイントテーブルから上記他の制御経路の制御経路番号と一致するエントリを無効化する。そしてさらに、ステップ S 2 5 では、例外抑止データブレーク履歴テーブルにおいて、該他の制御経路の制御経路番号に一致するエントリを無効化する。

【 0 0 7 3 】

次に、ステップ S 2 6 でデータブレーク処理を実行し、ステップ S 2 7 においては、例外抑止データブレーク履歴テーブルから実行が確定した制御経路の制御経路番号に一致するエントリを削除する。そして、上記ステップ S 1 4 へ進む。

【 0 0 7 4 】

一方、ステップ S 4 0 ではソフトウェアブレーク処理が実行され、ステップ S 1 4 へ進む。

【 0 0 7 5 】

以上より、本実施の形態 1 に係る計算機によれば、上記のような動作を実現するソフトウェアを実行することにより、投機的に移動した命令のうち本来の順序

による実行では確定していない命令によりデータブレイクが生じてプログラムの実行が中断してしまうことが回避されるため、データ処理能力及び動作の信頼性が高い計算機を得ることができる。

#### 【実施の形態 2】

図 16 は、本発明の実施の形態 2 に係る計算機の構成を示す図である。図 16 に示されるように、本実施の形態 2 に係る計算機は、図 6 に示された実施の形態 1 に係る計算機と同様な構成を有するが、履歴制御部 219 をさらに含むと共に、命令実行部 335 には例外抑止履歴確認制御部 26 と、例外抑止フラグ無効化命令実行部 217 と、例外抑止履歴読み出し命令実行部 28 と、例外抑止履歴書き込み命令実行部 20 とを含む点で相違する。また、コミット例外割り込み制御部 44 を含む割り込み制御部 10 を備える点で相違する。

#### 【0076】

さらに、命令実行部 335 に含まれるデータブレイク検出部 333 の構成は、後に詳しく説明するように、従来のデータブレイク検出部 305 の構成と相違する。

#### 【0077】

ここで、履歴制御部 219 は例外抑止履歴書き込み命令実行部 20 と例外抑止ロード命令実行部 24、例外抑止履歴確認制御部 26、例外抑止履歴読み出し命令実行部 28、例外抑止フラグ無効化命令実行部 217 及びコミット例外割り込み制御部 44 に接続される。また、例外抑止履歴書き込み命令実行部 20 と例外抑止履歴確認制御部 26、例外抑止履歴読み出し命令実行部 28、及び例外抑止フラグ無効化命令実行部 217 は、さらに命令デコード部 17 と汎用レジスタ 37 に接続される。

#### 【0078】

また、コミット例外割り込み制御部 44 はさらにプログラムカウンタ (PC) 13 と、レジスタ 31, 33, 35 に接続される。そして、データブレイク検出部 333 は、さらに命令デコード部 17 に接続される。

#### 【0079】

ここで、例外抑止履歴確認制御部 26 は、コミット命令が供給された場合には

、該命令により指定された汎用レジスタ 3 7 内のレジスタに対応する例外抑止フラグ 3 8 が有効か否か確認する。そして、該フラグに 1 が設定されており有効であると判断された場合には、該例外抑止フラグ 3 8 を 0 にして無効とし、履歴制御部 2 1 9 へ確認信号が供給される。これにより、コミット例外が発生したことが割り込み制御部 1 0 へ伝達される。なお、例外抑止フラグ 3 8 が無効であると判断された場合には、コミット命令による動作を終了する。

#### 【 0 0 8 0 】

次に、例外抑止履歴読み出し命令実行部 2 8 は、例外抑止履歴読み出し命令が供給された場合、履歴制御部 2 1 9 へ読み出し信号を供給することにより、履歴制御部 2 1 9 に記憶されている上記例外情報を読み出し、汎用レジスタ 3 7 へ書き込む。また、同様に例外抑止履歴書き込み命令実行部 2 0 は、例外抑止履歴書き込み命令が供給された場合、汎用レジスタ 3 7 から読み出したデータと書き込み信号を履歴制御部 2 1 9 へ供給して、該読み出したデータを履歴制御部 2 1 9 に書き込む。さらに、例外抑止フラグ無効化命令実行部 2 1 7 は例外抑止フラグ無効化命令が供給された場合には、汎用レジスタ 3 7 の指定されたレジスタ番号に対応する例外抑止フラグ 3 8 を 0 とすると共に、履歴制御部 2 1 9 へ無効化信号を供給して、指定されたレジスタに対応する例外情報を無効化する。

#### 【 0 0 8 1 】

また、コミット例外割り込み制御部 4 4 は、履歴制御部 2 1 9 から供給された割り込みを通知するコミット信号 CM に応じて、割り込みからの復帰先の命令アドレスをレジスタ (EPCR) 3 1 へ書き込み、割り込み発生前の動作状態を示すデータをレジスタ (EPSR) 3 3 へ書き込み、発生した割り込みに対応する動作状態をレジスタ (PSR) 3 5 へ書き込む。また、発生した割り込みに対応する分岐先アドレスをプログラムカウンタ (PC) 1 3 へ供給する。

#### 【 0 0 8 2 】

次に履歴制御部 2 1 9 は、例外抑止ロード命令実行部 2 4 から供給された登録信号 ADD に応じて、例外抑止履歴テーブルに例外情報を登録する。また、履歴制御部 2 1 9 は、例外抑止履歴確認制御部 2 6 から供給された確認信号 CC に応じて、指定されたレジスタ番号に例外情報が保持されているか否か確認する。そ



して、指定されたレジスタ番号に例外情報が保持されている場合には、コミット例外を検出したことを示すコミット信号CMをコミット例外割り込み制御部44へ供給する。

## 【0083】

また履歴制御部219は、例外抑止履歴読み出し命令実行部28から供給された読み出し信号Rに応じて、記憶されている例外情報を読み出して例外抑止履歴読み出し命令実行部28へ供給する。さらに、例外抑止履歴書き込み命令実行部20から供給された書き込み信号Wに応じて、供給されたデータを記憶する。また、例外抑止フラグ無効化命令実行部217から供給されたりセット信号RSに応じて、指定されたレジスタ番号の例外情報が保持されている場合に、そのエントリを無効化する。

## 【0084】

図17は、図16に示されたデータブレーク検出部333の構成を示す図である。図17に示されるように、本実施の形態2に係るデータブレーク検出部333は、図3に示された従来のデータブレーク検出部305と同様な構成を有するが、命令デコード部17に接続された例外抑止判定部337をさらに備える点で相違する。また、各NEフィールド338～341が、対応するVフィールド323～326に並設され、AND回路343～346を含む点で相違する。ここで、各AND回路343～346の入力端は、対応する検出部311～314及び例外抑止判定部337に接続され、出力端は対応するNEフィールド338～341に接続される。

## 【0085】

ここで、上記各NEフィールド338～341は、例外抑止命令に対してデータブレークを検出したか否かを示し、データブレークポイントレジスタを構成する。そして、NEフィールド338～341に0が書き込まれたときには例外抑止命令に対してデータブレークが未検出であることを示し、1が書き込まれたときには例外抑止命令に対してデータブレークが検出済みであることを示す。

## 【0086】

また、例外抑止判定部337は、ブレーク対象命令が例外抑止命令か否かを判

定し、データブ레이크条件が成立した場合には、対応するNEフィールド338～341にその出力値が書き込まれる。

#### 【0087】

図18は、図16に示された履歴制御部219の構成を示す図である。図18に示されるように、履歴制御部219は、アドレスレジスタ57と、データタイプレジスタ59と、レジスタ番号レジスタ61と、例外要因（EC）レジスタ175と、デコード回路65と、比較器67～69と、ECフィールド（EC）177～179と、Vフィールド151～153と、アドレスフィールド（ADDR）71, 75, 79と、データタイプフィールド（DT）72, 76, 80と、レジスタ番号フィールド（REG #）73, 77, 81と、コミット判定部180と、コミットエントリ検出部181と、無効エントリ検出部87と、例外抑止履歴登録制御部183と、例外抑止履歴確認制御部185と、例外抑止履歴読み出し命令実行部187と、例外抑止履歴書き込み命令実行部189と、履歴テーブル無効化部218を含む。

#### 【0088】

ここでアドレスレジスタ57と、データタイプレジスタ59と、レジスタ番号レジスタ61と、ECレジスタ175及びデコード回路65は命令実行部335に接続される。そして、アドレスレジスタ57は、例外抑止ロード命令が実行される際の実効アドレスを保持する。また、データタイプレジスタ59は、例外抑止ロード命令の実行においてロードストア対象とされるデータの型（大きさ）を示す識別値を保持する。そして、レジスタ番号レジスタ61は、例外抑止ロード命令の実行において書き込み対象とするレジスタ番号を保持する。

#### 【0089】

そして、ECレジスタ175は例外抑止ロード命令を実行するとき検出された例外要因の識別値を保持する。ここで、該例外要因とその識別値の例は、上記の表1に示されたものとなる。

#### 【0090】

また、デコード回路65は命令実行部335から供給された信号を解析し、対応する制御部を起動する。すなわち、供給される登録信号に応じて例外抑止履歴

登録制御部 1 8 3 を起動し、確認信号に応じて例外抑止履歴確認制御部 1 8 5 を起動し、読み出し信号に応じて例外抑止履歴読み出し命令実行部 1 8 7 を起動し、書き込み信号に応じて例外抑止履歴書き込み命令実行部 1 8 9 を起動し、無効化信号に応じて履歴テーブル無効化部 2 1 8 を起動する。

## 【 0 0 9 1 】

一方比較器 6 7 ～ 6 9 は、レジスタ番号レジスタ 6 1 及び対応するエントリに接続される。ここで、並設された複数のエントリは例外抑止履歴テーブルを構成するが、各エントリは例外要因を示す EC フィールド (EC) 1 7 7 ～ 1 7 9 と、対応するエントリに例外が発生したか否かを 2 値データにより示す V フィールド 1 5 1 ～ 1 5 3 と、例外処理の対象とするデータの実効アドレスを示すアドレスフィールド (ADDR) 7 1, 7 5, 7 9 と、ロード操作の対象とされるデータの型を示すデータタイプフィールド (DT) 7 2, 7 6, 8 0 と、例外情報が書き込まれたレジスタの番号を示すレジスタ番号フィールド (REG #) 7 3, 7 7, 8 1 とを含む。

## 【 0 0 9 2 】

なお、上記データタイプフィールド (DT) 7 2, 7 6, 8 0 にはデータの型に応じた識別値が記録されるが、この識別値の例は以下の表 2 に示される。

## 【 0 0 9 3 】

【表 2】

データ型	識別値
符合なしバイト	0
符合付きバイト	1
符合なしハーフワード	2
符合付きハーフワード	3
ワード	4
ダブルワード	5
クワッドワード	6

上記表 2 に示されるように、該識別値は例えば符号無しバイトに対して 0、符号付きバイトに対しては 1、符号無しハーフワードには 2、符号付きハーフワードには 3、ワードには 4、ダブルワードには 5、クワッドワードには 6 とされる。

## 【0 0 9 4】

そして上記比較器 6 7 ～ 6 9 は、EC フィールド (EC) 1 7 7 ～ 1 7 9 が有効なエントリに対して、レジスタ番号フィールド 7 3, 7 7, 8 1 に記録された値から特定される例外情報が書き込まれたレジスタと、コミット命令により指定されたレジスタとが一致しているか否かを比較し、両者が一致するか否かを示す信号を出力する。

## 【0 0 9 5】

また、コミット判定部 1 8 0 は比較器 6 7 ～ 6 9 に接続され、比較器 6 7 ～ 6 9 から供給された信号に応じて、例外情報が格納されているレジスタがコミット命令により指定されているか否かを判定する。そして、その判定結果を例外抑止履歴確認制御部 1 8 5 へ出力すると共に、コミット信号 CM をコミット例外割り込み制御部 4 4 へ供給する。

## 【0 0 9 6】

そして、コミットエントリ検出部 1 8 1 は、比較器 6 7 ～ 6 9 に接続され、比較器 6 7 ～ 6 9 から供給された信号に応じて、コミット命令により指定されたレジスタの番号と所定のフィールドに記録されているレジスタの番号とが一致したエントリの番号を検出する。

## 【 0 0 9 7 】

また、無効エントリ検出部 8 7 は、各エントリの EC フィールド 1 7 7, 1 7 8, 1 7 9 の情報に応じて、空きエントリ（無効なエントリ）の検出を行う。また、例外抑止履歴登録制御部 1 8 3 は、アドレスレジスタ 5 7 とデータタイプレジスタ 5 9、レジスタ番号レジスタ 6 1、EC レジスタ 1 7 5、デコード回路 6 5 及び無効エントリ検出部 8 7 に接続され、デコード回路 6 5 から供給された登録信号 ADD に応じて、無効エントリ検出部 8 7 で検出された空きエントリの EC フィールド 1 7 7, 1 7 8, 1 7 9 と、アドレスフィールド (ADDR) 7 1, 7 5, 7 9 と、レジスタ番号フィールド (REG #) 7 3, 7 7, 8 1 と、データタイプフィールド (DT) 7 2, 7 6, 8 0 へ例外情報を書き込む。

## 【 0 0 9 8 】

また、例外抑止履歴確認制御部 1 8 5 は、レジスタ番号レジスタ 6 1 と、コミット判定部 1 8 0 と、コミットエントリ検出部 1 8 1 と、デコード回路 6 5 及び V フィールド 1 5 1 ～ 1 5 3 に接続される。そして、例外抑止履歴確認制御部 1 8 5 は、コミット判定部 1 8 0 においてコミット命令により指定されたレジスタの番号と所定のフィールドに記録されている結果書き込み対象のレジスタの番号とが一致したと判定された場合、一致した該レジスタ番号が格納されているレジスタの V フィールド 1 5 1 ～ 1 5 3 へ 1 を書き込む。

## 【 0 0 9 9 】

また、例外抑止履歴読み出し命令実行部 1 8 7 は、指定されたエントリから例外情報を読み出して命令実行部 3 3 5 へ供給する。さらに、例外抑止履歴書き込み命令実行部 1 8 9 は、命令実行部 3 3 5 から EC レジスタ 1 7 5 やレジスタ番号レジスタ 6 1、アドレスレジスタ 5 7、及びデータタイプレジスタ 5 9 を介して供給された値を、例外抑止履歴書き込み命令で指定されたエントリへ書き込む。また、履歴テーブル無効化部 2 1 8 は、デコード回路 6 5 から供給された例

外抑止フラグ無効化命令において指定された該エントリの E C フィールド 1 7 7 , 1 7 8 , 1 7 9 を無効化する。

#### 【 0 1 0 0 】

以下において、図 1 6 に示された計算機の、割り込み処理プログラムによるデータブレーク割り込み動作を、図 1 9 のフローチャートを参照しつつ説明する。なお、図 1 6 に示された本実施の形態 2 に係る計算機においては、上記実施の形態 1 に係る計算機と同様に、メモリ 1 内の所定のアドレス領域に例外抑止ロード命令テーブルと、コミットポイントテーブル、コミットブレークポイントテーブル、及び例外抑止データブレーク履歴テーブルが格納される。

#### 【 0 1 0 1 】

図 1 9 に示されるように、まずステップ S 1 ではコンテキストが退避される。そして次に、ステップ S 2 においてデータブレーク検出部 3 3 3 に含まれるデータブレークポイントレジスタの N E フィールド 3 3 8 ~ 3 4 1 を参照し、ブレーク対象命令が例外抑止命令であるか否かを確認する。そしてステップ S 3 において、N E フィールド 3 3 8 ~ 3 4 1 に 1 が書き込まれていることにより該命令が例外抑止命令であると判断された場合には、ステップ S 4 へ進み、N E フィールド 3 3 8 ~ 3 4 1 に 0 が書き込まれていることにより例外抑止命令でないと判断された場合にはステップ S 2 0 へ進む。

#### 【 0 1 0 2 】

このとき、ステップ S 4 では例外抑止ロード命令テーブルを参照し、該ブレーク対象命令が含まれる制御経路番号を求める。そして、ステップ S 5 において、メモリ 1 内に格納された上記例外抑止データブレーク履歴テーブルに、ブレーク対象命令が含まれる上記制御経路番号と、命令アドレスと、実効アドレスとを登録する。次に、ステップ S 6 において、上記制御経路番号に基づき、メモリ 1 に格納されたコミットポイントテーブルを参照し、他の制御経路が存在するか否かを確認する。

#### 【 0 1 0 3 】

そして、ステップ S 7 において演算命令実行部 2 2 の比較動作により他の制御経路が存在すると判断された場合にはステップ S 8 へ進み、存在しないと判断さ

れた場合にはステップ S 1 1 へ進む。ここでステップ S 8 では、見出された制御経路番号が求められ、ステップ S 9 ではコミットブレイクポイントテーブルへ、該他の制御経路のコミットポイントに対応するブレイクポイントの命令アドレスと、制御経路番号とを登録する。このようにして、ステップ S 1 0 において該他の制御経路のコミットポイントに対するブレイクポイントを設定する。

#### 【0104】

そして、ステップ S 1 1 において、ステップ S 1 で退避されたコンテキストが復元される。なお、ステップ S 2 0 ではデータブレイク処理を実行し、上記ステップ S 1 1 へ動作が進行する。そしてさらにステップ S 1 2 においては、割り込み復帰命令を実行することにより、データブレイク割込みの処理から元のプログラムの実行へ復帰し、割込み処理動作を終了する。

#### 【0105】

なお、図 1 6 に示された本実施の形態に係る計算機の、割込み処理プログラムによるソフトウェアブレイク割込み動作は、図 1 1 のフローチャートに示された実施の形態 1 に係る計算機による動作と同様である。

#### 【0106】

以上より、本実施の形態 2 に係る計算機によれば、履歴制御部 2 1 9 は例外抑止履歴テーブルに例外情報を登録し、データブレイク検出部 3 3 3 は N E フィールド 3 3 8 ~ 3 4 1 に該例外動作を留保する（例外抑止命令である）ことを示す 1 のデータを書き込んだ上で、コミット命令の実行により該留保した例外動作を上記例外情報に応じて遂行するため、投機的に移動した命令のうち本来の順序による実行では確定していない命令によりデータブレイクが生じてプログラムの実行が中断してしまうことが回避され、データ処理能力及び動作の信頼性が高い計算機を得ることができる。

#### 〔実施の形態 3〕

図 2 0 は、本発明の実施の形態 3 に係る計算機の構成を示す図である。図 2 0 に示されるように、本実施の形態 3 に係る計算機は、図 1 6 に示された実施の形態 2 に係る計算機と同様な構成を有するが、ブレイク履歴制御部 3 5 5 をさらに含むと共に、命令実行部 3 5 3 にはブレイク履歴読み出し命令実行部 3 4 9 と、

ブレーク履歴書込み命令実行部 3 5 1 とを含む点で相違する。また後述するように、命令実行部 3 5 3 に含まれたデータブレーク検出部 3 4 7 の構成が、図 1 6 に示されたデータブレーク検出部 3 3 3 の構成と相違する。

#### 【 0 1 0 7 】

ここで、ブレーク履歴読み出し命令実行部 3 4 9 は命令デコード部 1 7 と、ブレーク履歴制御部 3 5 5 及び履歴制御部 2 1 9 に接続される。また、ブレーク履歴書込み命令実行部 3 5 1 は命令デコード部 1 7 と、汎用レジスタ 3 7、履歴制御部 2 1 9 及びブレーク履歴制御部 3 5 5 に接続される。また、ブレーク履歴制御部 3 5 5 は、さらに例外抑止履歴確認制御部 2 6 と例外抑止フラグ無効化命令実行部 2 1 7、データブレーク検出部 3 4 7 及びコミット例外割り込み制御部 4 4 に接続される。

#### 【 0 1 0 8 】

そして、例外抑止履歴確認制御部 2 6 は上記実施の形態 2 における場合の動作に加え、コミット命令の実行によりブレーク履歴制御部 3 5 5 に対して確認信号 CC を供給する。また、例外抑止フラグ無効化命令実行部 2 1 7 は上記実施の形態 2 における場合の動作に加え、例外抑止フラグ無効化命令の実行によりブレーク履歴制御部 3 5 5 に対してリセット（無効化）信号 RS を供給する。

#### 【 0 1 0 9 】

また、ブレーク履歴読み出し命令実行部 3 4 9 は、命令デコード部 1 7 からブレーク履歴読み出し命令が供給された場合、ブレーク履歴制御部 3 5 5 に対して読み出し信号 R を供給することにより例外抑止データブレーク履歴テーブルを読み出し、その結果を汎用レジスタ 3 7 へ書き込む。また、ブレーク履歴書込み命令実行部 3 5 1 は、命令デコード部 1 7 からブレーク履歴書込み命令が供給された場合、汎用レジスタ 3 7 から読み出したデータを書き込み信号 W と共にブレーク履歴制御部 3 5 5 へ供給する。

#### 【 0 1 1 0 】

そして、ブレーク履歴制御部 3 5 5 は、データブレーク検出部 3 4 7 から供給された登録信号 ADD に応じて、後述する例外抑止データブレーク履歴テーブル部にデータブレーク情報を登録する。また、供給された確認信号 CC に応じて、



指定されたレジスタ番号にデータブ레이크情報が保持されているか否か確認する。そして、指定されたレジスタ番号にデータブ레이크情報が保持されている場合には、ブ레이크割込みを検出した旨をブ레이크信号BRによりコミット例外割り込み制御部44へ伝達する。また、供給された無効化信号RSに応じて、指定されたレジスタ番号にデータブ레이크情報が保持されている場合には、そのエントリを無効化する。また、供給された読み出し信号Rに応じて例外抑止データブ레이크履歴テーブル部から読み出したデータをブ레이크履歴読み出し命令実行部349へ供給する。さらには、供給された書込み信号Wに応じて、ブ레이크履歴書き込み命令実行部351から供給された値を例外抑止データブ레이크履歴テーブル部へ書き込む。

## 【0111】

図21は、図20に示されたデータブ레이크検出部347の構成を示す図である。図21に示されるように、本実施の形態に係るデータブ레이크検出部347は、図17に示されたデータブ레이크検出部333と同様な構成を有するが、出力端がブ레이크履歴制御部355に接続されたAND回路348を備えている点で相違するものである。

## 【0112】

なお、図21に示された検出部311～314は、データブ레이크条件が成立したか否かを判定する。具体的には、非例外抑止命令であるロードストア命令の実効アドレスと、アドレスフィールドに格納されたアドレスとを比較し、それらが一致した場合には対応するVフィールドに1を書き込み、データブ레이크が成立した旨を示す信号mtを出力する。また、例外抑止命令の実効アドレスとアドレスフィールドに格納されたアドレスとを比較して一致した場合には、OR回路327を介して登録信号ADDをブ레이크履歴制御部355へ供給する。

## 【0113】

図22は、図20に示されたブ레이크履歴制御部355の構成を示す図である。図22に示されるように、本実施の形態に係るブ레이크履歴制御部355は、図18に示された履歴制御部219と同様な構成を有し、アドレスレジスタ57と、実効アドレスレジスタ359と、レジスタ番号レジスタ61と、有効レジス

タ 3 5 7 と、デコード回路 6 5 と、比較器 6 7 ~ 6 9 と、E フィールド (E) 3 6 1 ~ 3 6 3 と、V フィールド 1 5 1 ~ 1 5 3 と、アドレスフィールド (ADDR) 7 1, 7 5, 7 9 と、E A フィールド (EA) 3 6 4 ~ 3 6 6 と、レジスタ番号フィールド (REG #) 7 3, 7 7, 8 1 と、ブレーク判定部 3 6 7 と、ブレークエントリ検出部 3 6 9 と、無効エントリ検出部 8 7 と、ブレーク履歴登録制御部 1 8 3 b と、ブレーク履歴確認制御部 1 8 6 と、ブレーク履歴読み出し命令実行部 1 8 7 b と、ブレーク履歴書き込み命令実行部 1 8 9 b と、履歴テーブル無効化部 2 1 8 とを含む。

## 【 0 1 1 4 】

ここでアドレスレジスタ 5 7 と、実効アドレスレジスタ 3 5 9 と、レジスタ番号レジスタ 6 1 と、有効レジスタ 3 5 7 及びデコード回路 6 5 は命令実行部 3 5 3 に接続される。そして、アドレスレジスタ 5 7 は、ブレーク対象命令の命令アドレスを保持する。また、実効アドレスレジスタ 3 5 9 は、ブレーク対象命令の実効アドレスを保持する。そして、レジスタ番号レジスタ 6 1 は、例外抑止ロード命令の実行において書き込み対象とするレジスタ番号を保持する。

## 【 0 1 1 5 】

また、有効レジスタ 3 5 7 は有効フラグからなるものとすることができ、データブレーク情報が有効か否かを示すデータを保持する。そして、0 が書き込まれているときにはエントリが無効であることを意味し、1 が書き込まれているときにはエントリが有効であることを示す。

## 【 0 1 1 6 】

また、デコード回路 6 5 は命令実行部 3 5 3 から供給された信号を解析し、対応する制御部を起動する。すなわち、供給される登録信号 ADD に応じてブレーク履歴登録制御部 1 8 3 b を起動し、確認信号 CC に応じてブレーク履歴確認制御部 1 8 6 を起動し、読み出し信号 R に応じてブレーク履歴読み出し命令実行部 1 8 7 b を起動し、書き込み信号 W に応じてブレーク履歴書き込み命令実行部 1 8 9 b を起動し、無効化信号 RS に応じて履歴テーブル無効化部 2 1 8 を起動する。

## 【 0 1 1 7 】

一方比較器 6 7 ~ 6 9 は、レジスタ番号レジスタ 6 1 及び対応するエントリに接続される。ここで、並設された複数のエントリは例外抑止データブレイク履歴テーブルを構成するが、各エントリはエントリが有効か否かを 2 値データにより示す E フィールド (E) 3 6 1 ~ 3 6 3 と、対応するエントリにデータブレイクが発生したか否かを 2 値データにより示す V フィールド 1 5 1 ~ 1 5 3 と、登録されているデータブレイクの命令アドレスを保持するアドレスフィールド (ADDR) 7 1, 7 5, 7 9 と、登録されているデータブレイクの実効アドレスを保持する EA フィールド 3 6 4 ~ 3 6 6 と、登録されているデータブレイクの結果を書き込むレジスタの番号を示すレジスタ番号フィールド (REG #) 7 3, 7 7, 8 1 とを含む。

## 【 0 1 1 8 】

そして上記比較器 6 7 ~ 6 9 は、E フィールド (E) 3 6 1 ~ 3 6 3 が有効なエントリに対して、レジスタ番号フィールド 7 3, 7 7, 8 1 に記録された値と、レジスタ番号レジスタ 6 1 に保持されたレジスタ番号とを比較し、両者が一致するか否かを示す信号を出力する。

## 【 0 1 1 9 】

また、ブレイク判定部 3 6 7 は比較器 6 7 ~ 6 9 に接続され、比較器 6 7 ~ 6 9 から供給された信号に応じて、データブレイク情報が格納されているレジスタがコミット命令により指定されているか否かを判定する。そして、その判定結果をブレイク履歴確認制御部 1 8 6 へ出力すると共に、ブレイク信号 B R をコミット例外割り込み制御部 4 4 へ供給する。

## 【 0 1 2 0 】

そしてブレイクエントリ検出部 3 6 9 は、比較器 6 7 ~ 6 9 に接続され、比較器 6 7 ~ 6 9 から供給された信号に応じて、例外抑止データブレイク履歴に登録されている結果書込み対象を指定するレジスタ番号のうちコミット命令により指定されたレジスタの番号を格納するエントリ番号を検出する。また、無効エントリ検出部 8 7 は、各エントリの E フィールド 3 6 1 ~ 3 6 3 の情報に応じて、例外抑止データブレイク履歴テーブルにおける空きエントリ (無効なエントリ) の検出を行う。

## 【 0 1 2 1 】

また、ブレーク履歴登録制御部 1 8 3 b は、アドレスレジスタ 5 7 と実効アドレスレジスタ 3 5 9、レジスタ番号レジスタ 6 1、有効レジスタ 3 5 7、デコード回路 6 5 及び無効エントリ検出部 8 7 に接続され、デコード回路 6 5 から供給された登録信号 A D D に応じて、無効エントリ検出部 8 7 で検出された空きエントリの E フィールド 3 6 1 ~ 3 6 3 と、アドレスフィールド (A D D R) 7 1, 7 5, 7 9 と、レジスタ番号フィールド (R E G #) 7 3, 7 7, 8 1 と、E A フィールド 3 6 4 ~ 3 6 6 へデータブレーク情報を書き込む。

## 【 0 1 2 2 】

また、ブレーク履歴確認制御部 1 8 6 は、レジスタ番号レジスタ 6 1 と、ブレーク判定部 3 6 7 と、ブレークエントリ検出部 3 6 9 と、デコード回路 6 5 及び V フィールド 1 5 1 ~ 1 5 3 に接続される。そして、ブレーク履歴確認制御部 1 8 6 は、ブレーク判定部 3 6 7 においてコミット命令により指定されたレジスタの番号と例外抑止データブレーク履歴に登録されている結果書込み対象を指定するレジスタ番号とが一致したと判定された場合、一致した該レジスタ番号が格納されているエントリの V フィールド 1 5 1 ~ 1 5 3 へ 1 を書き込む。

## 【 0 1 2 3 】

また、ブレーク履歴読み出し命令実行部 1 8 7 b は、例外抑止データブレーク履歴読み出し命令により指定されたエントリからデータブレーク情報を読み出して命令実行部 3 5 3 へ供給する。さらに、ブレーク履歴書き込み命令実行部 1 8 9 b は、命令実行部 3 5 3 から有効レジスタ 3 5 7 やレジスタ番号レジスタ 6 1、アドレスレジスタ 5 7、及び実効アドレスレジスタ 3 5 9 を介して供給された値を、例外抑止データブレーク履歴書き込み命令で指定されたエントリへ書き込む。また、履歴テーブル無効化部 2 1 8 は、デコード回路 6 5 から供給された例外抑止フラグ無効化命令において指定された該エントリの E フィールド 3 6 1 ~ 3 6 3 を無効化する。

## 【 0 1 2 4 】

以上より、本実施の形態 3 に係る計算機によれば、上記実施の形態 2 に係る計算機と同様な効果を奏すると共に、ブレーク履歴制御部 3 5 5 は実行を留保する

データブ레이크についてのデータブ레이크情報を例外抑止データブ레이크履歴テーブルに登録した上で、コミット命令の実行により該留保したデータブ레이크動作を上記データブ레이크情報に応じて遂行するため、データ処理に必要なプログラムをさらに短縮化することができる。従って、該プログラムの実行に必要とされるメモリ容量が低減できると共に、データ処理をより高速化することができる。

#### 【実施の形態 4】

図 2 3 は、本発明の実施の形態 4 に係る計算機の構成を示す図である。図 2 3 に示されるように、本実施の形態に係る計算機は、図 1 6 に示された実施の形態 2 に係る計算機と同様な構成を有するが、命令実行部 3 7 3 にはモード命令実行部 3 7 1 がさらに含まれ、レジスタ制御部 3 7 7 にはモードレジスタ 3 7 5 がさらに含まれる点で相違するものである。

#### 【0 1 2 5】

ここで、このモード命令実行部 3 7 1 は命令デコード部 1 7 と、モードレジスタ 3 7 5 及び汎用レジスタ 3 7 に接続される。また、モードレジスタ 3 7 5 は、さらに例外抑止履歴確認制御部 2 6 と例外抑止フラグ無効化命令実行部 2 1 7 に接続される。

#### 【0 1 2 6】

そして、以上のような構成を有する本実施の形態 4 に係る計算機においては、本来の命令のみを実行するときにはモードレジスタ 3 7 5 に 0 の値が格納され、本来の命令に加えてコミット命令あるいは例外抑止フラグ無効化命令が実行されたことに応じて割り込みを発生させるときにはモードレジスタ 3 7 5 に 1 の値が格納される。

#### 【0 1 2 7】

また、例外抑止履歴確認制御部 2 6 は、モードレジスタ 3 7 5 に 1 の値が格納されているときに限り、割込信号を出力することによってコミット命令の実行による割り込みの発生を割り込み制御部 1 0 へ通知する。また、例外抑止フラグ無効化命令実行部 2 1 7 は、モードレジスタ 3 7 5 に 1 の値が格納されているときに限り、割込信号を出力することによって例外抑止フラグ無効化命令の実行によ

る割り込みの発生を割り込み制御部 1 0 へ通知する。

#### 【 0 1 2 8 】

また、命令実行回路 2 3 は、命令デコード部 1 7 から動作モード読み出し命令が供給された場合には、モードレジスタ 3 7 5 から格納されている値を読み出し、その値を汎用レジスタ 3 7 へ書き込む。また、同様に動作モード書き込み命令が供給された場合には、汎用レジスタ 3 7 から読み出した値をモードレジスタ 3 7 5 へ書き込む。

#### 【 0 1 2 9 】

なお、本実施の形態 4 に係る計算機においては、メモリ 1 内の所定のアドレス領域に、例外抑止データブレーク履歴テーブルが格納される。そして、本実施の形態 4 に係る例外抑止データブレーク履歴テーブルは、図 2 5 に示されるように、ブレーク対象とされる命令の実行結果を書き込むレジスタの番号と、該ブレーク対象とされる命令の命令アドレスと、上記命令の実効アドレスとを組にしたデータ、すなわち (r 0, a 0, e a 0)、(r 1, a 1, e a 1)、… (r 1, a 1, e a 1) から構成される。

#### 【 0 1 3 0 】

以下において、図 2 3 に示された計算機の、割り込み処理プログラムによるデータブレーク割り込み動作を、図 2 4 のフローチャートを参照しつつ説明する。

#### 【 0 1 3 1 】

なお、図 2 3 に示された本実施の形態 4 に係る計算機の、割り込み処理プログラムによるソフトウェアブレーク割り込み動作は、図 5 のフローチャートに示された従来の計算機による動作と同様である。

#### 【 0 1 3 2 】

図 2 4 に示されるように、まずステップ S 1 ではコンテキストを退避する。次にステップ S 2 では割り込みの種類を判定し、ステップ S 3 では該割り込みがコミット命令または例外抑止フラグ無効化命令の実行に起因するものであるか否かが判断される。そして、該割り込みがコミット命令または例外抑止フラグ無効化命令の実行のいずれにも起因しない場合にはステップ S 3 0 へ進み、いずれかに起因する場合にはステップ S 2 0 へ進む。さらにステップ S 2 0 では、コミット

命令の実行に起因するものであるか否かが判断され、コミット命令の実行に起因するものである場合にはステップ S 4 へ進み、コミット命令でなく例外抑止フラグ無効化命令の実行に起因する場合にはステップ S 2 1 へ進む。

【0 1 3 3】

そして、ステップ S 4 ではメモリ 1 に格納された例外抑止データブレイク履歴テーブル内に、コミット命令により指定されたレジスタ番号が実行結果を書き込むレジスタの番号データ r 1 として格納されているか否か確認する。そして、ステップ S 5 において該レジスタ番号が存在しないと判断したときには、ステップ S 8 へ進む一方、該レジスタ番号が存在すると判断したときには、ステップ S 6 へ進んでそのレジスタ番号により指定されるレジスタを結果書き込みの対象とする該データブレイク処理を実行する。その後、ステップ S 7 では上記例外抑止データブレイク履歴テーブルより、コミット命令により指定されたレジスタ番号が含まれる一組（エントリ）のデータを削除する。

【0 1 3 4】

次に、ステップ S 8 では例外抑止データブレイク履歴テーブルに有効なエントリが存在するか否か確認し、ステップ S 9 において存在しないと判断された場合にはステップ S 1 0 へ進み、存在すると判断された場合にはステップ S 1 1 へ進む。ここで、ステップ S 1 0 ではモード書き込み命令を実行することにより、モードレジスタ 3 7 5 に 0 の値が格納される。

【0 1 3 5】

そして、ステップ S 1 1 においてコンテキストの復元がなされ、ステップ S 1 2 において割り込み復帰命令を実行することにより、割り込み処理から復帰し、データブレイク割り込みの動作を終了する。

【0 1 3 6】

一方、ステップ S 2 1 では、メモリ 1 に格納された例外抑止データブレイク履歴テーブル内に、例外抑止フラグ無効化命令により指定されたレジスタ番号が実行結果を書き込むレジスタの番号データ r 1 として格納されているか否か確認する。そして、ステップ S 2 2 において該レジスタ番号が存在しないと判断したときには、ステップ S 2 4 へ進む一方、該レジスタ番号が存在すると判断したとき

には、ステップ S 2 3 へ進んで上記例外抑止データブ레이크履歴テーブルより、例外抑止フラグ無効化命令により指定されたレジスタ番号が含まれる一組（エントリ）のデータを削除する。

#### 【 0 1 3 7 】

次に、ステップ S 2 4 では例外抑止データブ레이크履歴テーブルに有効なエントリが存在するか否か確認し、ステップ S 2 5 において存在しないと判断された場合にはステップ S 2 6 へ進み、存在すると判断された場合にはステップ S 1 1 へ進む。ここで、ステップ S 2 6 ではモード書込み命令を実行することによってモードレジスタ 3 7 5 に 0 の値が格納され、ステップ S 1 1 へ進む。

#### 【 0 1 3 8 】

一方、ステップ S 3 0 においては、読み出したブ레이크対象命令が例外抑止命令であるか否かが、演算命令実行部 2 2 の命令比較動作により判定される。そして、ステップ S 3 1 において該命令が例外抑止命令であると判断された場合には、ステップ S 3 2 へ進み、例外抑止命令でないと判断された場合にはステップ S 4 0 へ進む。

#### 【 0 1 3 9 】

そして、ステップ S 3 2 において、メモリ 1 内に格納された上記例外抑止データブ레이크履歴テーブルに、ブ레이크対象命令の実行結果を書き込むレジスタの番号と、命令アドレスと、実効アドレスとを登録する。その後、ステップ S 3 3 においては、モードレジスタ 3 7 5 に格納された値が 0 であるか否か判断され、0 であると判断された場合にはステップ S 3 4 へ進み、1 であると判断された場合にはステップ S 1 1 へ進む。ここで、ステップ S 3 4 においては、モード書込み命令を実行することによりモードレジスタ 3 7 5 に 1 の値が格納され、ステップ S 1 1 へ進む。

#### 【 0 1 4 0 】

一方、ステップ S 4 0 においてはデータブ레이크処理が実行され、ステップ S 1 1 へ進む。

#### 【 0 1 4 1 】

以上より、本実施の形態 4 に係る計算機によれば、例外抑止命令の実行におい



てデータブレイク処理の実行を留保（抑止）した場合には、モードレジスタ 3 7 5 に 1 の値を設定することとするため、モードレジスタ 3 7 5 に格納された値を参照することにより、確定した命令の実行において必要なデータブレイク処理の有無を迅速に判断でき、データ処理をさらに高速化することができる。

【 0 1 4 2 】

最後に、本発明の課題を解決するための手段について付記する。

（１）プログラムされた命令の実行を制御する計算機の制御方法であって、プログラムにおいて分岐命令より後置された命令を、分岐命令より先に実行する第一のステップと、第一のステップにおいて、プログラムの実行を中断する必要性が見出された場合に、中断を留保する第二のステップと、分岐命令を実行することにより選択された分岐先の命令の実行において、第二のステップで留保した中断が必要とされる場合には、中断を遂行する第三のステップとを有することを特徴とする計算機の制御方法。

（２）第二のステップにおいては、留保される中断を実行するための情報を記憶し、第三のステップにおいては、情報に応じて中断を遂行する（１）に記載の計算機の制御方法。また、留保される中断を実行するための情報を記憶し、上記情報に応じて上記中断を遂行することとすれば、所定の処理を遂行するためのプログラムを短縮化することができるため、該プログラムを記憶するために必要な記憶容量を低減して、動作の高速化と回路規模の低減を図ることができる。

（３）前記情報を無効化する第四のステップをさらに有する（２）に記載の計算機の制御方法。このように、必要に応じて上記情報を無効化することとすれば、分岐命令の実行の結果必要とされないことが確定した情報を削除することができるため、ハードウェア資源の有効活用を図ることができる。

（４）第二のステップでは、フラグを所定の値に設定し、第三のステップでは、分岐命令を実行することにより選択された分岐先の命令の実行において、第二のステップで留保した中断が必要とされるか否かをフラグの値を参照することにより判断する（１）に記載の計算機の制御方法。このように、分岐命令を実行することにより選択された分岐先の命令の実行において、中断を必要とするか否かをフラグの値を参照することにより判断すれば、該判断を迅速に行うことができる

ため、計算機の回路規模の増大を抑制しつつデータ処理の高速化を図ることができる。

(5) 所定の命令を実行することによりフラグの値を無効化するステップをさらに有する(4)に記載の計算機の制御方法。

(6) 第三のステップにおける分岐命令を実行することにより選択された分岐先の命令は、割り込み処理プログラムにより実行される(1)に記載の計算機の制御方法。

(7) プログラムされた命令を実行する計算機であって、分岐命令より後置された所定の命令を分岐命令より先に実行したとき、プログラムの実行を中断する必要性が見出された場合に中断を留保する例外抑止手段と、分岐命令を実行することにより選択された分岐先の命令の実行において、例外抑止手段により留保された中断が必要とされる場合には、中断を遂行する中断処理実行手段とを備えたことを特徴とする計算機。

(8) 留保される中断を実行するための情報を記憶する記憶手段をさらに備え、前記中断処理実行手段は、前記記憶手段に記憶された前記情報に応じて前記中断を遂行する(7)に記載の計算機。

(9) 情報を無効化する無効化手段をさらに備えた(8)に記載の計算機。

(10) フラグをさらに備え、例外抑止手段はフラグを所定の値に設定すると共に、中断処理実行手段は、分岐命令を実行することにより選択された分岐先の命令の実行において、先に留保した中断が必要とされるか否かをフラグの値を参照することにより判断する(7)に記載の計算機。

(11) 所定の命令を実行することによりフラグを無効化するフラグ無効化手段をさらに備えた(10)に記載の計算機。

(12) 分岐命令を実行することにより選択された分岐先の命令を、割り込み処理プログラムにより実行する割り込み処理実行手段をさらに備えた(7)に記載の計算機。

【0143】

【発明の効果】

上述の如く、プログラムにおいて分岐命令より後置された命令を、分岐命令よ

り先に実行したときに必要とされた中断を留保し、分岐命令を実行することにより選択された分岐先の命令の実行において、その留保した中断が必要とされる場合に該中断を遂行すれば、分岐命令より後置された命令を、分岐命令より先に実行することによる無用な中断を回避して、プログラムされた命令の実行を確実に遂行することができるため、計算機の動作の信頼性を高めることができる。

【 0 1 4 4 】

ここで、上記中断を遂行した場合には割り込み処理プログラムを実行すれば、該中断により、例えばデバッグ支援プログラムの実行などによる異なるシステムの制御ができるため、多様なデータ処理を実現することができる。

【図面の簡単な説明】

【図 1】

従来の計算機の構成を示す図である。

【図 2】

図 1 に示された命令ブレーク検出部の構成を示す図である。

【図 3】

図 1 に示されたデータブレーク検出部の構成を示す図である。

【図 4】

図 1 に示された計算機の、割り込み処理プログラムによるデータブレーク割り込み動作を示すフローチャートである。

【図 5】

図 1 に示された計算機の、割り込み処理プログラムによるソフトウェアブレーク割り込み動作を示すフローチャートである。

【図 6】

本発明の実施の形態 1 に係る計算機の構成を示す図である。

【図 7】

図 6 に示された演算命令実行部の構成を示す図である。

【図 8】

図 6 に示された計算機の、データブレークを検出した際の動作を示すフローチャートである。

【図 9】

図 6 に示された計算機の、本来の順序での命令の実行が確定した際の動作を示すフローチャートである。

【図 10】

図 6 に示された計算機の、データブレイク割込みにおける割込み処理プログラムによる動作を示すフローチャートである。

【図 11】

図 6 に示された計算機の、割込み処理プログラムによるソフトウェアブレイク割込み動作を示すフローチャートである。

【図 12】

本発明の実施の形態 1 に係る例外抑止ロード命令テーブルの構成を示す図である。

【図 13】

本発明の実施の形態 1 に係るコミットポイントテーブルの構成を示す図である。

【図 14】

本発明の実施の形態 1 に係るコミットブレイクポイントテーブルの構成を示す図である。

【図 15】

本発明の実施の形態 1 に係る例外抑止データブレイク履歴テーブルの構成を示す図である。

【図 16】

本発明の実施の形態 2 に係る計算機の構成を示す図である。

【図 17】

図 16 に示されたデータブレイク検出部の構成を示す図である。

【図 18】

図 16 に示された履歴制御部の構成を示す図である。

【図 19】

図 16 に示された計算機の、割込み処理プログラムによるデータブレイク割込

み動作を示すフローチャートである。

【図 2 0】

本発明の実施の形態 3 に係る計算機の構成を示す図である。

【図 2 1】

図 2 0 に示されたデータブレーク検出部の構成を示す図である。

【図 2 2】

図 2 0 に示されたブレーク履歴制御部の構成を示す図である。

【図 2 3】

本発明の実施の形態 4 に係る計算機の構成を示す図である。

【図 2 4】

図 2 3 に示された計算機の、割込み処理プログラムによるデータブレーク割込み動作を示すフローチャートである。

【図 2 5】

本発明の実施の形態 4 に係る例外抑止データブレーク履歴テーブルの構成を示す図である。

【符号の説明】

- 1   メモリ
- 9, 10   割り込み制御部
- 11   命令読み出し制御部
- 13   プログラムカウンタ (PC)
- 15   命令語レジスタ (IR)
- 17   命令デコード部
- 19   ロード命令実行部
- 20   例外抑止履歴書き込み命令実行部
- 21   ストア命令実行部
- 22, 235   演算命令実行部
- 23   命令実行回路
- 24, 237   例外抑止ロード命令実行部
- 26   例外抑止履歴確認制御部

- 2 8 例外抑止履歴読み出し命令実行部
- 3 1 レジスタ (E P C R)
- 3 3 レジスタ (E P S R)
- 3 5 レジスタ (P S R)
- 3 7 汎用レジスタ
- 3 8 例外抑止フラグ
- 4 0, 5 5 割込制御回路
- 4 4 コミット例外割り込み制御部
- 5 7 アドレスレジスタ
- 5 9 データタイプレジスタ
- 6 1 レジスタ番号レジスタ
- 6 5 デコード回路
- 6 7 ~ 6 9 比較器
- 7 1, 7 5, 7 9, 3 1 5 ~ 3 1 8 アドレスフィールド (ADDR)
- 7 2, 7 6, 8 0 データタイプフィールド (DT)
- 7 3, 7 7, 8 1 レジスタ番号フィールド (REG #)
- 8 7 無効エントリ検出部
- 1 7 7 ~ 1 7 9 E C フィールド
- 1 8 0 コミット判定部
- 1 8 1 コミットエントリ検出部
- 1 8 3 例外抑止履歴登録制御部
- 1 8 3 b ブレーク履歴登録制御部
- 1 8 5 例外抑止履歴確認制御部
- 1 8 6 ブレーク履歴確認制御部
- 1 8 7 例外抑止履歴読み出し命令実行部
- 1 8 7 b ブレーク履歴読み出し命令実行部
- 1 8 9 例外抑止履歴書き込み命令実行部
- 1 8 9 b ブレーク履歴書き込み命令実行部
- 2 1 7 例外抑止フラグ無効化命令実行部

2 1 8 履歴テーブル無効化部  
 2 1 9 履歴制御部  
 2 4 8, 3 2 7 O R 回路  
 3 0 1 命令ブレーク検出部  
 3 0 3 命令読み出し部  
 3 0 5, 3 3 3, 3 4 7 データブレーク検出部  
 3 0 7, 3 2 9, 3 3 5, 3 5 3, 3 7 3 命令実行部  
 3 0 9, 3 3 1, 3 7 7 レジスタ制御部  
 3 1 1 ~ 3 1 4 検出部  
 3 1 9 ~ 3 2 2, 3 6 1 ~ 3 6 3 E フィールド ( E )  
 3 2 3 ~ 3 2 6 V フィールド  
 3 3 7 例外抑止判定部  
 3 3 8 ~ 3 4 1 N E フィールド  
 3 4 3 ~ 3 4 6, 3 4 8 A N D 回路  
 3 4 9 ブレーク履歴読み出し命令実行部  
 3 5 1 ブレーク履歴書込み命令実行部  
 3 5 5 ブレーク履歴制御部  
 3 5 7 有効レジスタ  
 3 5 9 実効アドレスレジスタ  
 3 6 4 ~ 3 6 6 実効アドレス ( E A ) フィールド  
 3 6 7 ブレーク判定部  
 3 6 9 ブレークエントリ検出部  
 3 7 1 モード命令実行部  
 3 7 5 モードレジスタ  
 A D D 登録信号  
 C C 確認信号  
 R 読み出し信号  
 W 書き込み信号  
 C M コミット信号

R S リセット信号

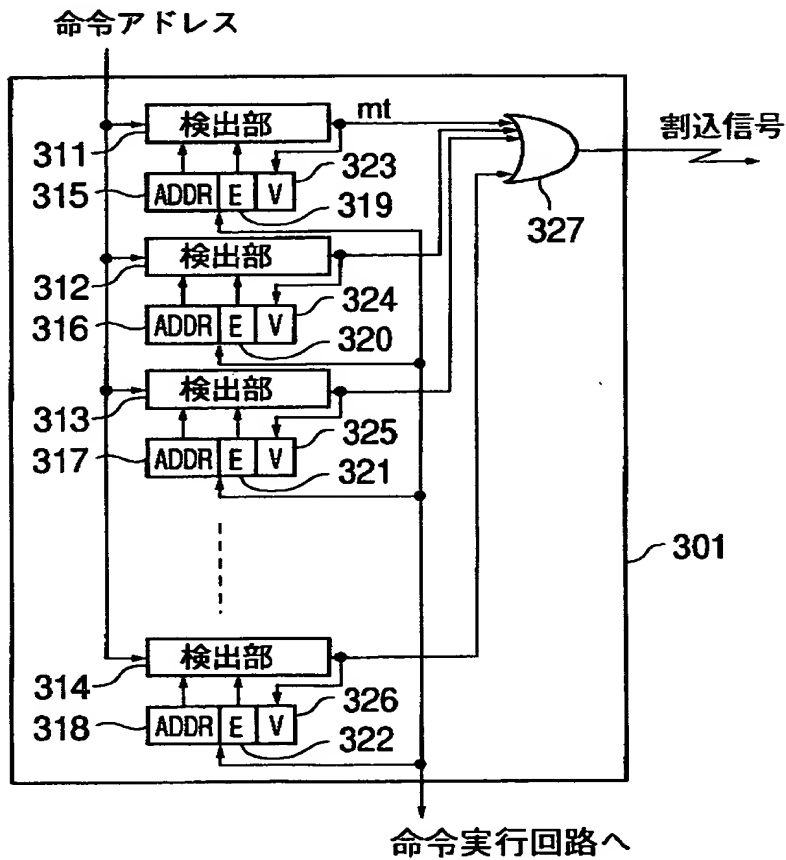
B R ブレーク信号





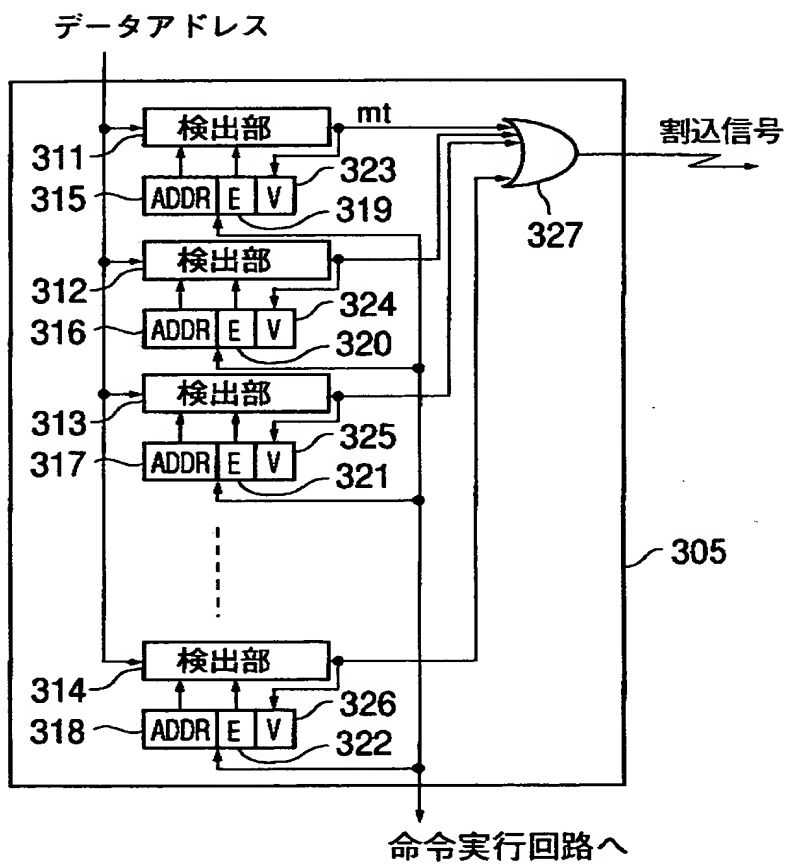
【図 2】

図1に示された命令ブレーク検出部の構成を示す図



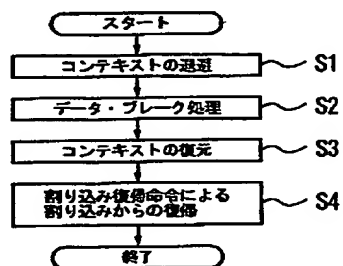
【図 3】

図1に示されたデータブレーク検出部の構成を示す図



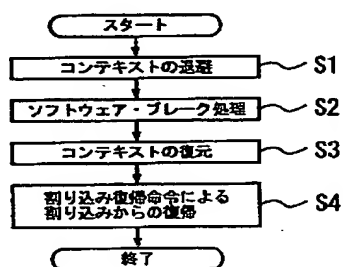
【図 4】

図1に示された計算機の、割込み処理プログラムによる  
データブレイク割込み動作を示すフローチャート



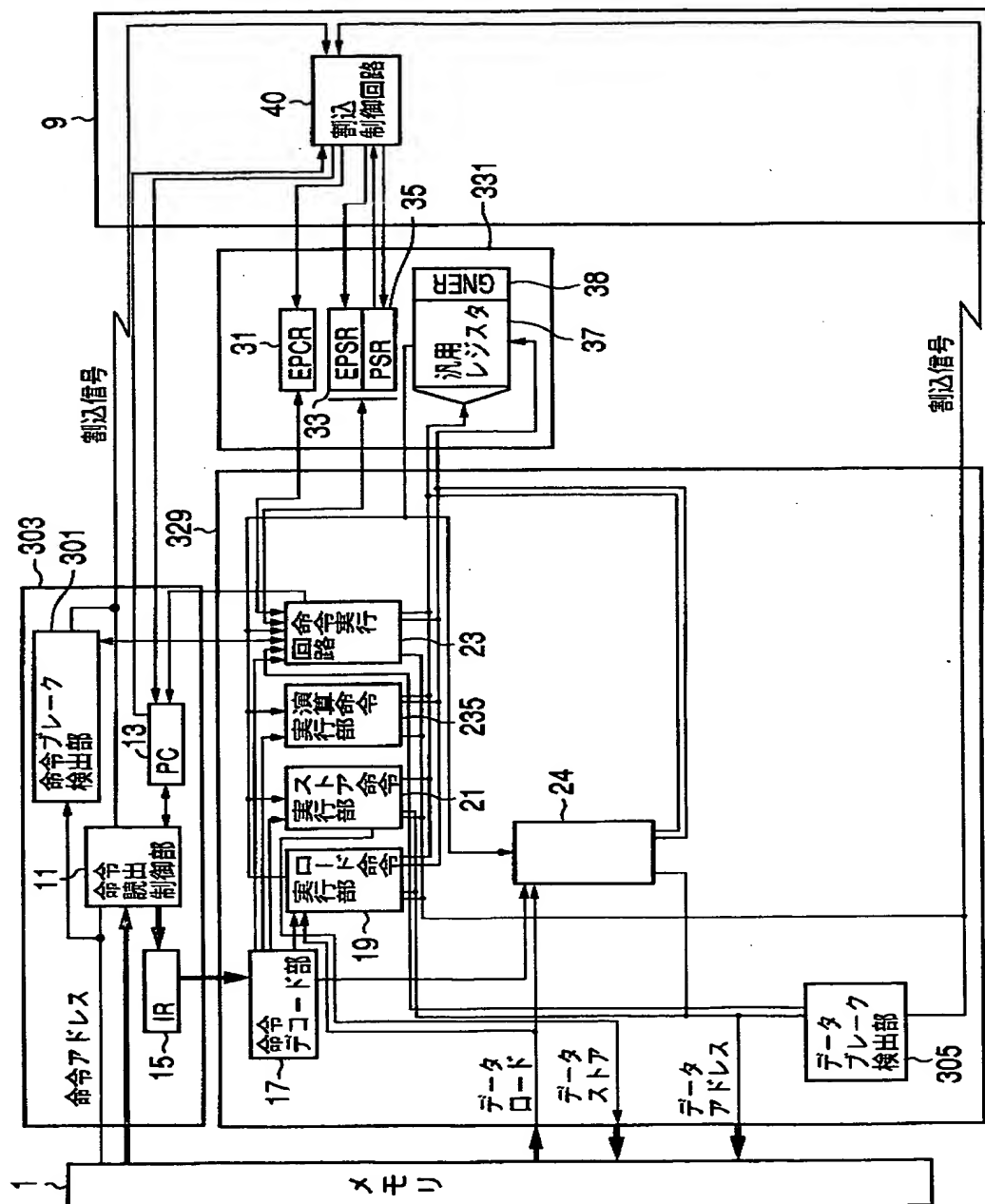
【図 5】

図1に示された計算機の、割込み処理プログラムによる  
ソフトウェアブレイク割込み動作を示すフローチャート



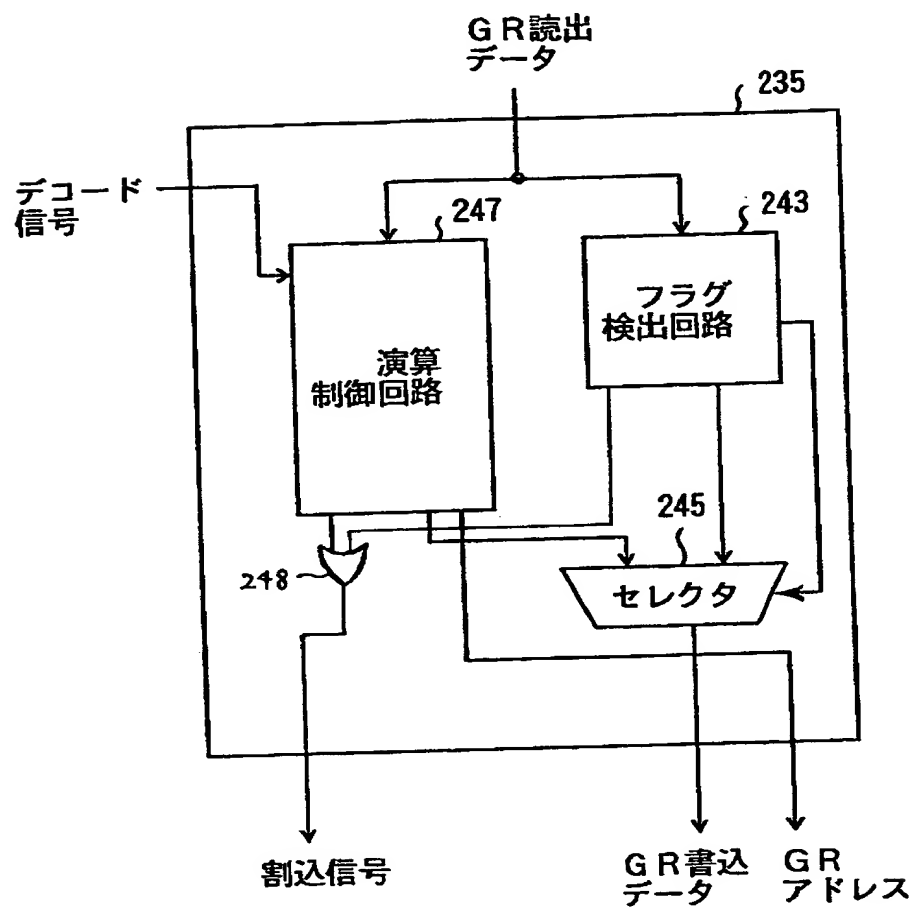
【図 6】

本発明の実施の形態1に係る計算機の構成を示す図



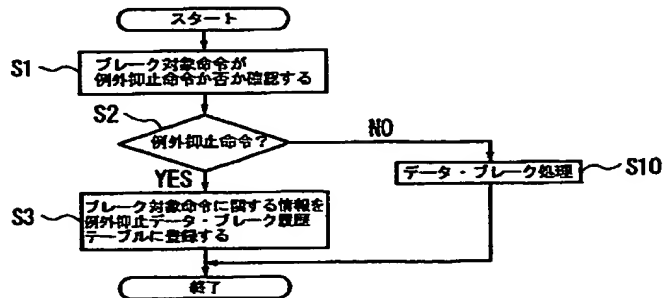
【図 7】

図 6 に示された演算命令実行部の構成を示す図



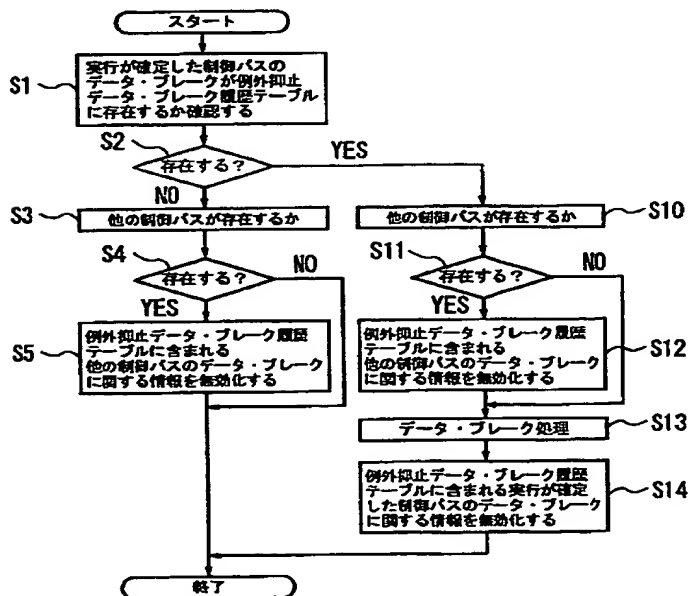
【図 8】

図6に示された計算機の、データブレイクを検出した際の動作を示すフローチャート



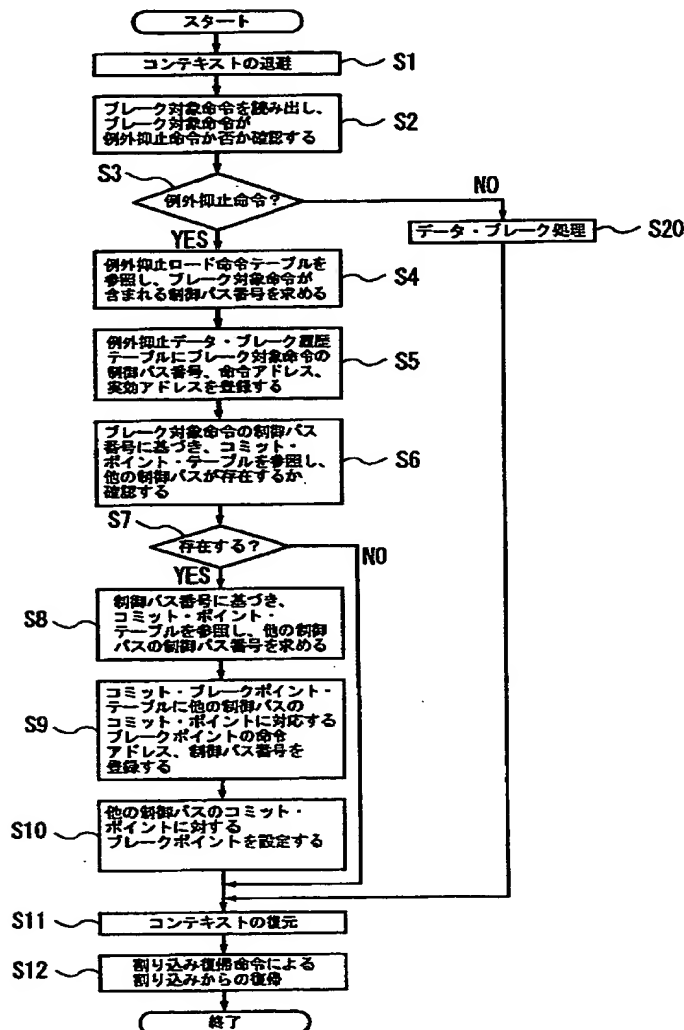
【図 9】

図6に示された計算機の、本来の順序での命令の実行が確定した際の動作を示すフローチャート



【図 10】

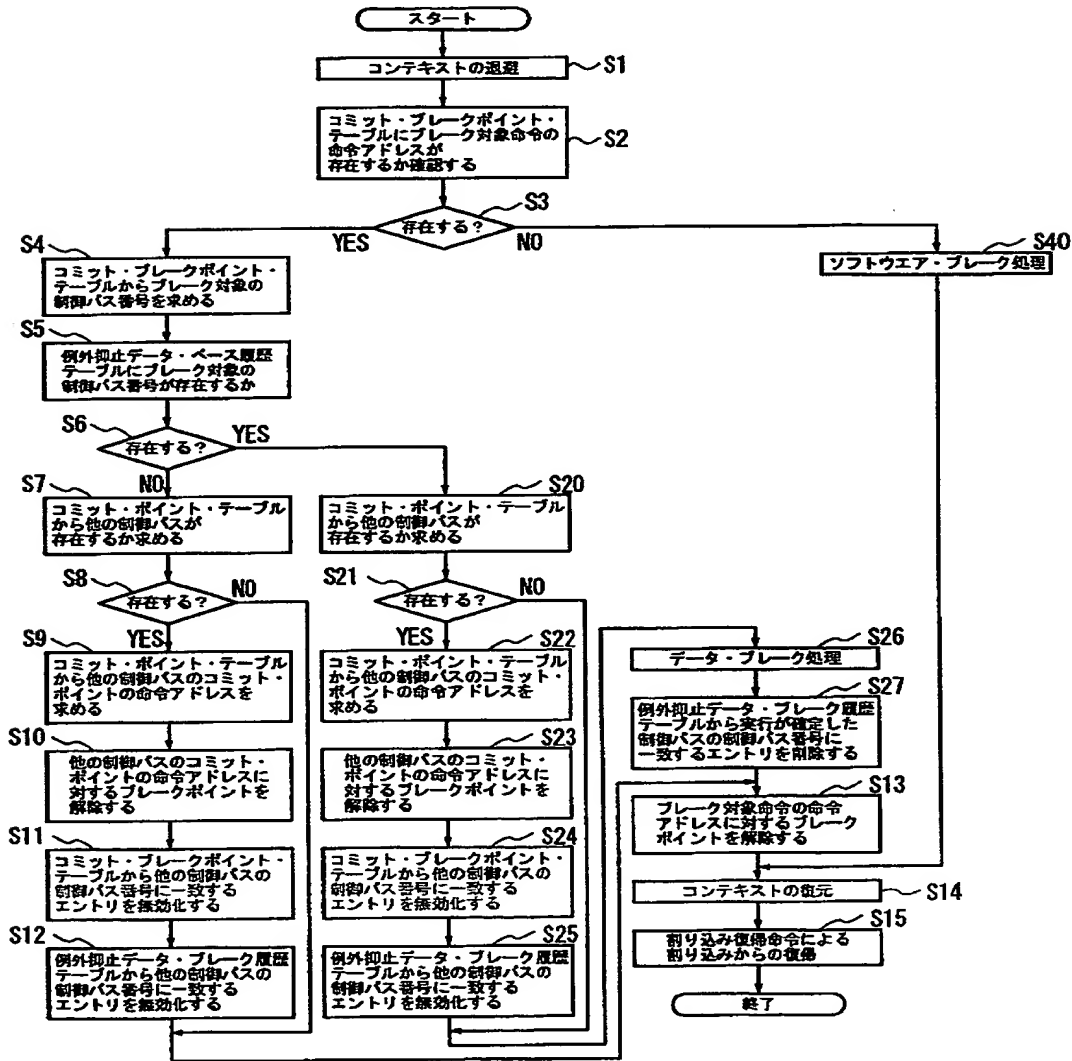
図6に示された計算機の、データブレイク割込みにおける割込み処理プログラムによる動作を示すフローチャート





【図 11】

図6に示された計算機の、割込み処理プログラムによるソフトウェアブレーク割込み動作を示すフローチャート



【図 1 2】

本発明の実施の形態1に係る例外抑止ロード  
命令テーブルの構成を示す図

命令アドレス	経路識別番号
a0	p0
a1	p1
⋮	⋮
⋮	⋮
ai	pi

【図 1 3】

本発明の実施の形態1に係るコミットポイント  
テーブルの構成を示す図

経路識別番号	命令アドレス	他の経路識別番号
p0	b0	sp0
p1	b1	sp1
⋮	⋮	⋮
⋮	⋮	⋮
pj	bj	spj

【図 1 4】

本発明の実施の形態1に係るコミットブレイク  
ポイントテーブルの構成を示す図

命令アドレス	経路識別番号
b0	p0
b1	p1
⋮	⋮
⋮	⋮
bk	pk

【図 1 5】

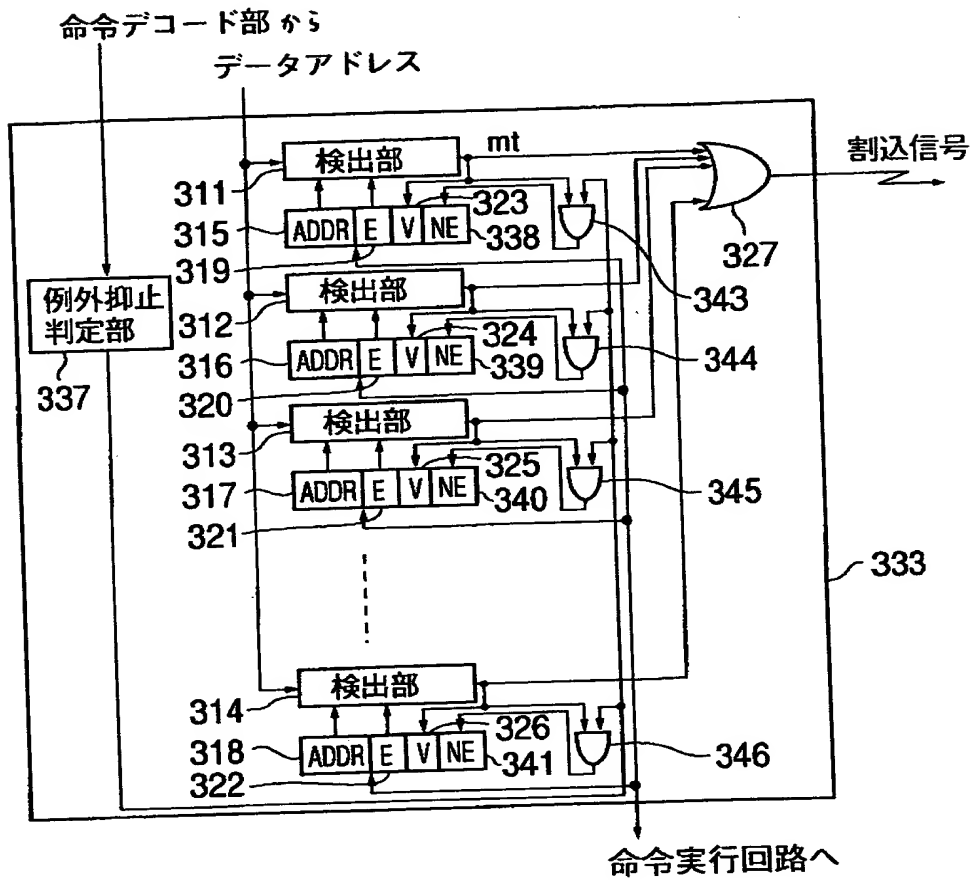
本発明の実施の形態1に係る例外抑止データブレイク履歴  
テーブルの構成を示す図

経路識別番号	命令アドレス	実効アドレス
p0	c0	ec0
p1	c1	ec1
⋮	⋮	⋮
⋮	⋮	⋮
pl	cl	ec1



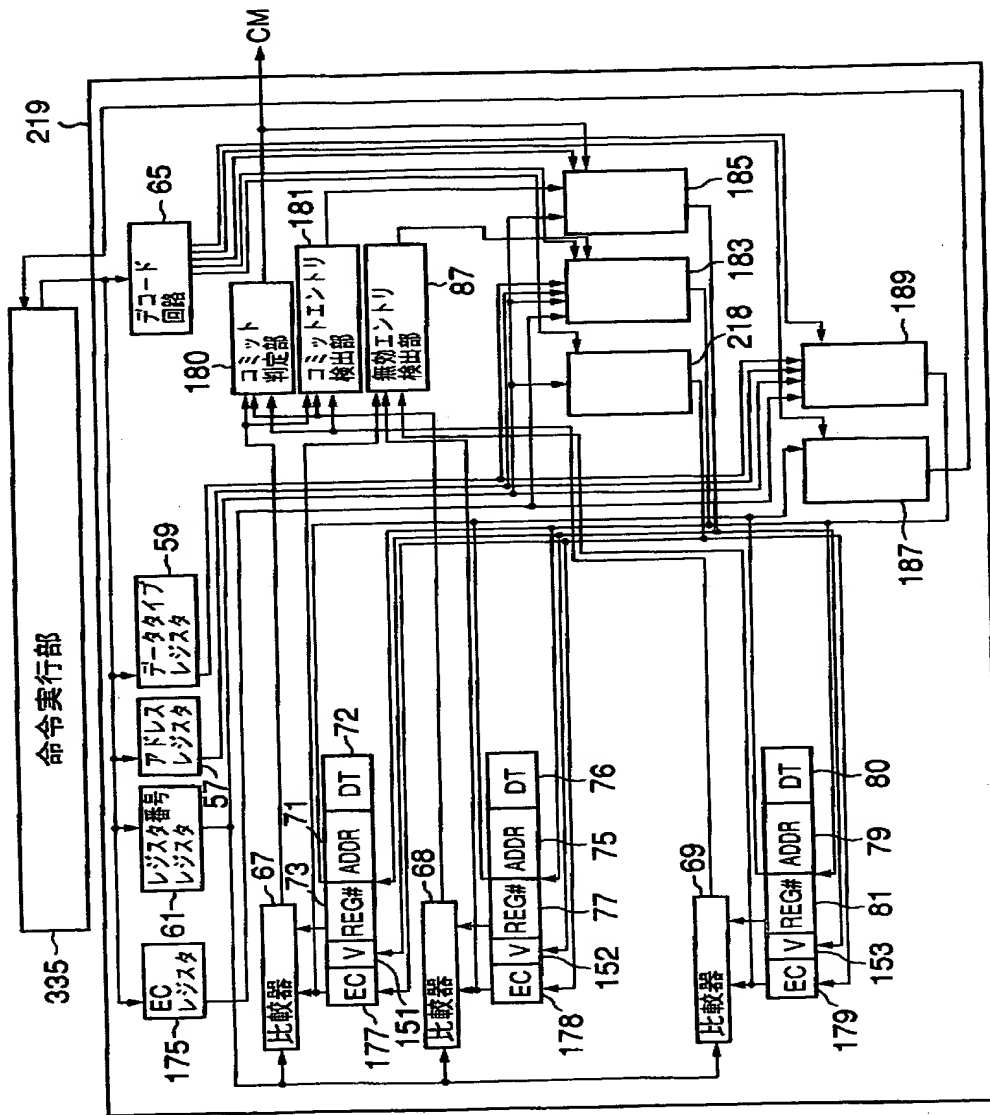
【図 1 7】

図16に示されたデータブレイク検出部の構成を示す図



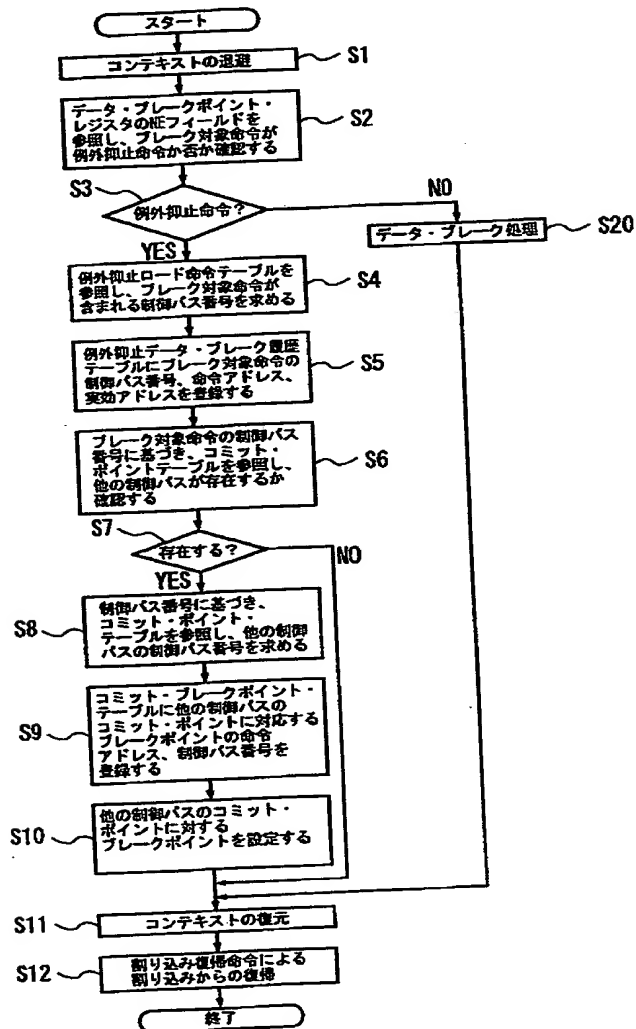
【図18】

図16に示された履歴制御部の構成を示す図



【図 19】

図 16 に示された 計算機 の、割込み処理プログラムによる  
データブレイク割込み動作を示すフローチャート

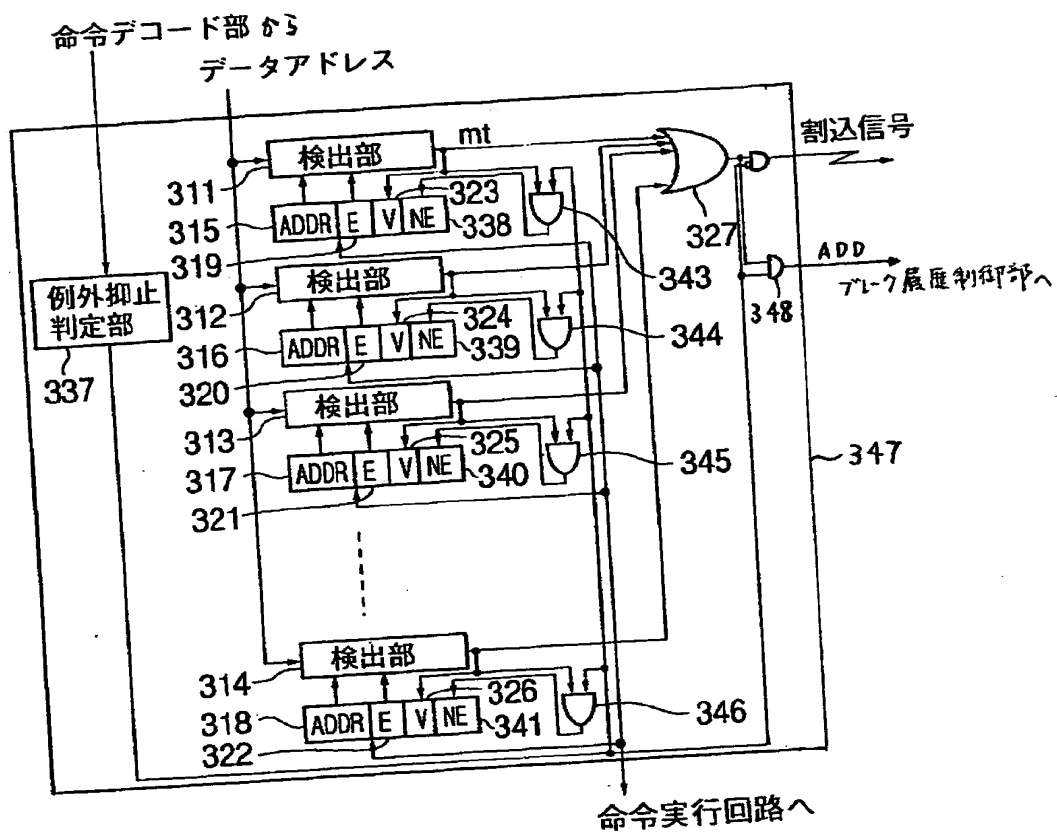






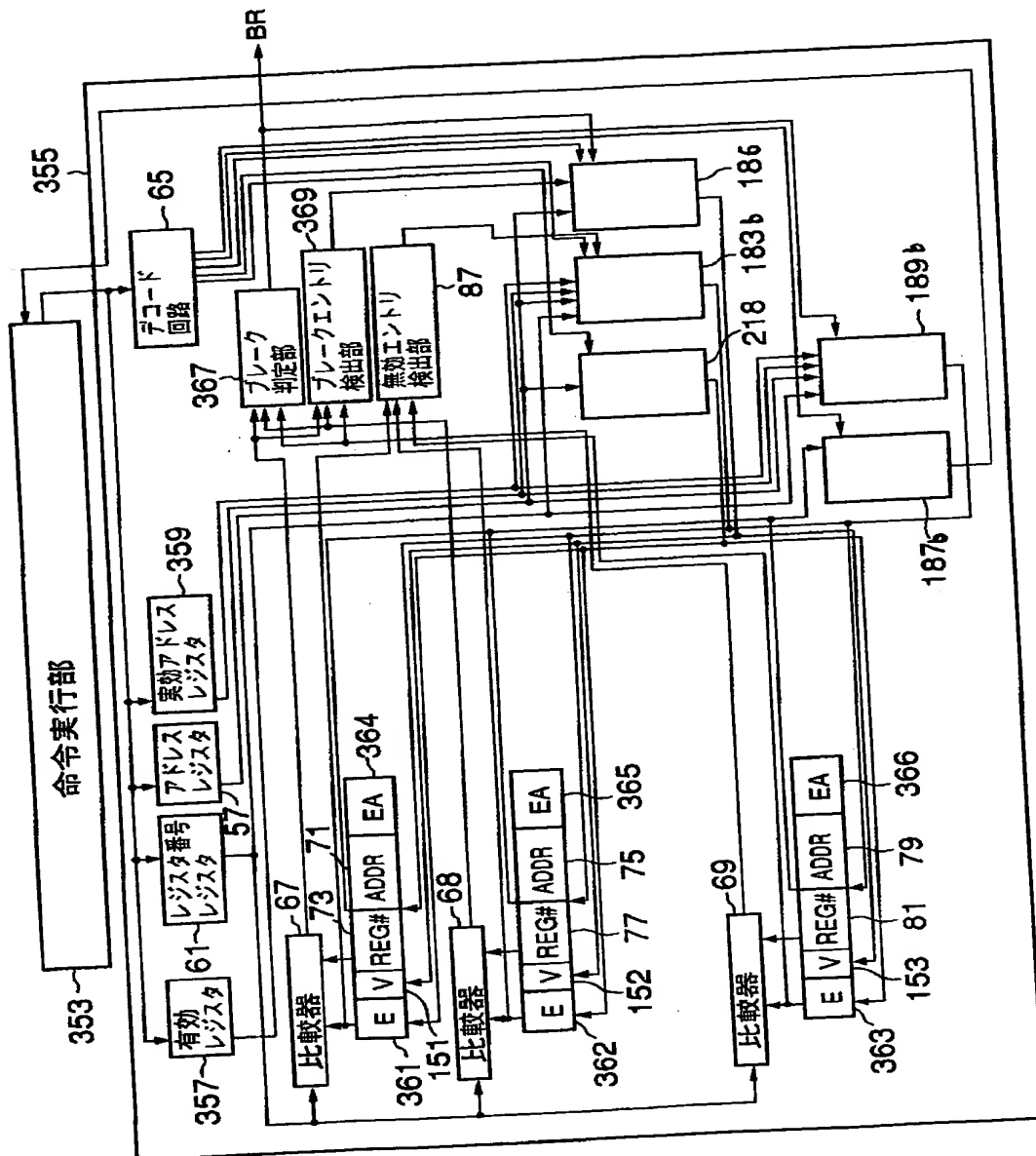
【図 21】

図 20 に示されたデータブレイク検出部の構成を示す図



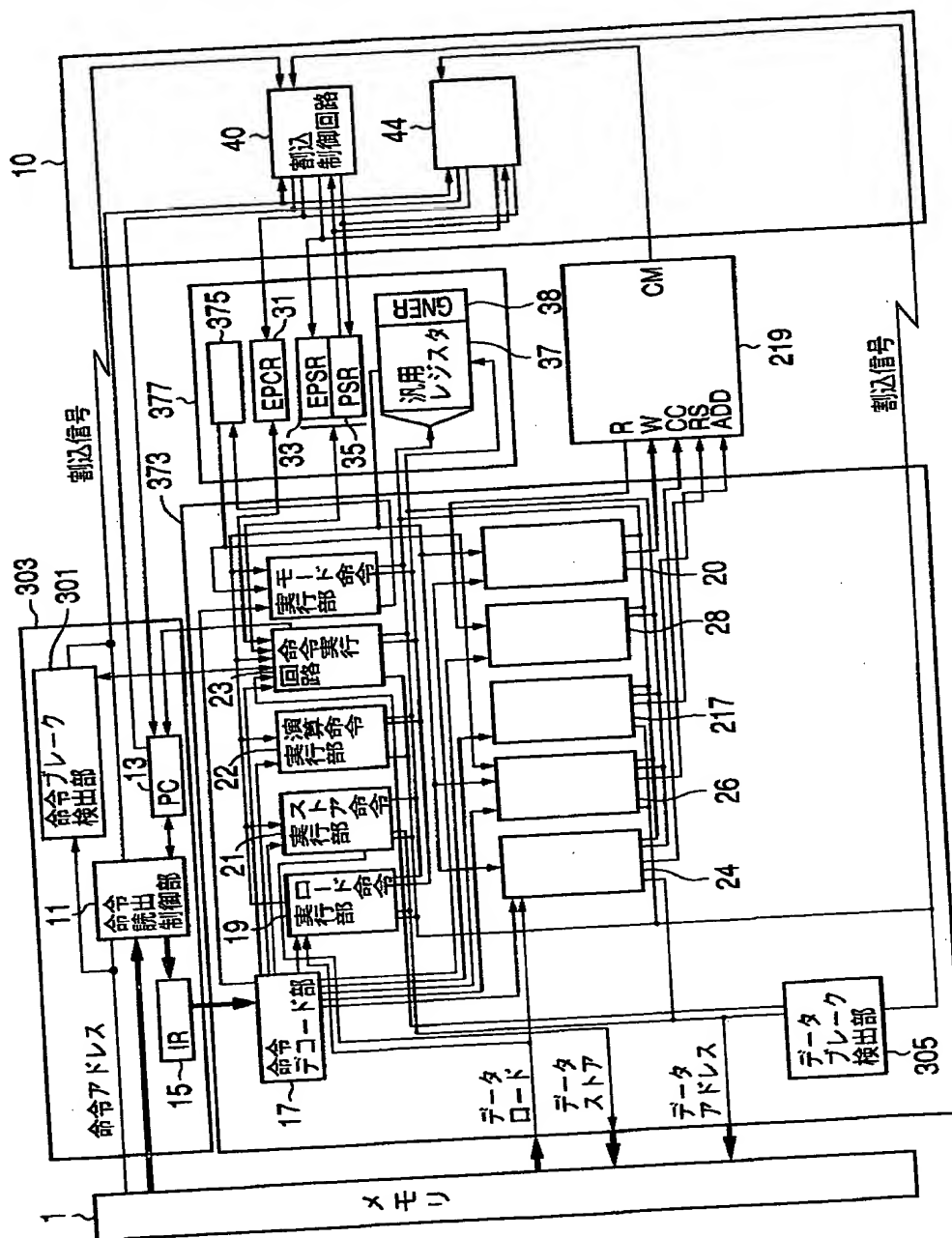
【図 2 2】

図20に示されたブレーク履歴制御部の構成を示す図



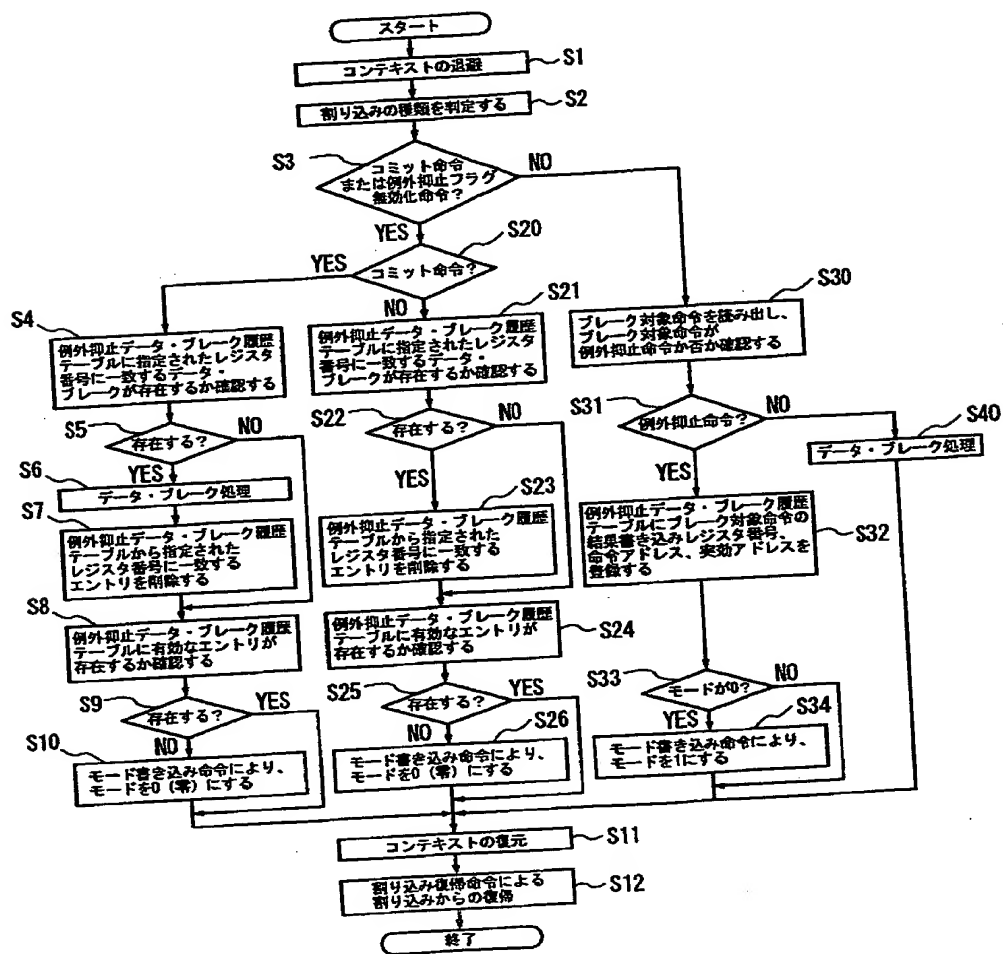
【圖 23】

本発明の実施の形態4に係る計算機の構成を示す図



【図24】

図23に示された計算機の、割り込み処理プログラムによるデータブレーク割り込み動作を示すフローチャート



【図 2 5】

本発明の実施の形態4に係る例外抑止データブレイク履歴  
テーブルの構成を示す図

レジスタ番号	命令アドレス	実効アドレス
r0	a0	ea0
r1	a1	ea1
⋮	⋮	⋮
⋮	⋮	⋮
rl	al	eal

【書類名】 要約書

【要約】

【課題】 データ処理能力が高く動作の信頼性が向上された計算機とその制御方法を提供する。

【解決手段】 プログラムされた命令を実行する計算機であって、分岐命令より後置された所定の命令を分岐命令より先に実行したとき、プログラムの実行を中断する必要性が見出された場合には、上記中断を留保する例外抑止ロード命令実行部24と、分岐命令を実行することにより選択された分岐先の命令の実行において、留保された中断の実行が必要とされる場合には、上記中断を遂行する割り込み制御部9とを備えたことを特徴とする計算機を提供する。

【選択図】 図6

特2000-067789

出 願 人 履 歴 情 報

識別番号

[000005223]

1. 変更年月日	1996年 3月26日
[変更理由]	住所変更
住 所	神奈川県川崎市中原区上小田中4丁目1番1号
氏 名	富士通株式会社